

# ON DETERMINING THE POWER OF A TEST AFTER DATA COLLECTION

by

WILLIAM AVRAM CHERNOFF

B.S., Chapman University, 2007

A REPORT

Submitted in partial fulfillment of the  
requirements for the degree

MASTER OF SCIENCE

Department of Statistics  
College of Art and Sciences

KANSAS STATE UNIVERSITY  
Manhattan, Kansas  
2009

Approved by

---

Major Professor  
Dr. Leigh Murray

# ABSTRACT

The term retrospective power describes methods for estimating the true power of a test after data have been collected. These methods have been recommended by some authors when null hypothesis of a test cannot be rejected. This report uses simulations to study power as a construct of an observed effect, variance, sample size, and set level of significance under the balanced one-way analysis of variance model for normally distributed populations with constant variance.

Retrospective power, as a construct of sample data, is not recommended when the null hypothesis of a test cannot be rejected. When the p-value of the test is large, estimates for true power tend to fall below the 0.80 level and width-minimized confidence limits for true power tend to be wide.

# TABLE OF CONTENTS

<b>List of Figures .....</b>	<b>v</b>
<b>List of Tables .....</b>	<b>vii</b>
<b>Acknowledgements .....</b>	<b>viii</b>
<b>1. Introduction .....</b>	<b>1</b>
<b>2. Literature Review .....</b>	<b>2</b>
<b>3. Methods .....</b>	<b>4</b>
<i>3.1 Model .....</i>	<i>4</i>
<i>3.2 Estimated Power .....</i>	<i>8</i>
<i>3.3 Confidence Interval Optimization .....</i>	<i>9</i>
<i>3.4 Parameter Settings .....</i>	<i>14</i>
<i>3.5 Iteration .....</i>	<i>16</i>
<b>4. Results .....</b>	<b>18</b>
<i>4.1 Two-Sample Symmetric Case .....</i>	<i>19</i>
<i>4.2 Three-Sample Symmetric Case .....</i>	<i>21</i>
<i>4.3 Three-Sample Asymmetric Case .....</i>	<i>23</i>
<i>4.4 Four Sample Symmetric Case A .....</i>	<i>25</i>
<i>4.5 Four Sample Symmetric Case B .....</i>	<i>27</i>
<i>4.6 Four Sample Asymmetric Case .....</i>	<i>29</i>

4.7 Symmetric Cases .....	31
4.8 Asymmetric Cases .....	31
4.9 Symmetric Cases versus Asymmetric Cases .....	32
<b>5. Conclusion .....</b>	<b>33</b>
<b>References .....</b>	<b>38</b>
<b>Appendix A .....</b>	<b>39</b>
<b>Appendix B .....</b>	<b>91</b>
<b>Appendix C .....</b>	<b>141</b>
<b>Appendix D .....</b>	<b>243</b>



# LIST OF FIGURES

Figure 3.1.a: Update Bracket Interval for Minimand for $v$ less than $u$ .....	12
Figure 3.1.b: Update Bracket Interval for Minimand for $v$ less than $u$ .....	12
Figure 4.1.a: Two-Sample Symmetric Case for Power vs Sample Size with Sigma 2.5 .....	20
Figure 4.1.b: Two-Sample Symmetric Case for Power vs Sample Size with Sigma 5.0 .....	20
Figure 4.1.c: Two-Sample Symmetric Case for Power vs Sample Size with Sigma 10.0 .....	20
Figure 4.1.d: Two-Sample Symmetric Case for Power vs Sample Size with Sigma 20.0 .....	20
Figure 4.2.a: Three-Sample Symmetric Case for Power vs Sample Size with Sigma 2.5 .....	22
Figure 4.2.b: Three-Sample Symmetric Case for Power vs Sample Size with Sigma 5.0 .....	22
Figure 4.2.c: Three-Sample Symmetric Case for Power vs Sample Size with Sigma 10.0 .....	22
Figure 4.2.d: Three-Sample Symmetric Case for Power vs Sample Size with Sigma 20.0 .....	22
Figure 4.3.a: Three-Sample Asymmetric Case for Power vs Sample Size with Sigma 2.5 .....	24
Figure 4.3.b: Three-Sample Asymmetric Case for Power vs Sample Size with Sigma 5.0 .....	24
Figure 4.3.c: Three-Sample Asymmetric Case for Power vs Sample Size with Sigma 10.0 .....	24
Figure 4.3.d: Three-Sample Asymmetric Case for Power vs Sample Size with Sigma 20.0 .....	24
Figure 4.4.a: Four-Sample Symmetric Case A for Power vs Sample Size with Sigma 2.5 .....	26
Figure 4.4.b: Four-Sample Symmetric Case A for Power vs Sample Size with Sigma 5.0 .....	26
Figure 4.4.b: Four-Sample Symmetric Case A for Power vs Sample Size with Sigma 10.0 .....	26
Figure 4.4.c: Four-Sample Symmetric Case A for Power vs Sample Size with Sigma 20.0 .....	26
Figure 4.5.a: Four-Sample Symmetric Case B for Power vs Sample Size with Sigma 2.5 .....	28
Figure 4.5.a: Four-Sample Symmetric Case B for Power vs Sample Size with Sigma 5.0 .....	28
Figure 4.5.a: Four-Sample Symmetric Case B for Power vs Sample Size with Sigma 10.0 .....	28
Figure 4.5.a: Four-Sample Symmetric Case B for Power vs Sample Size with Sigma 20.0 .....	28
Figure 4.6.a: Four-Sample Asymmetric Case for Power vs Sample Size with Sigma 2.5 .....	30

Figure 4.6.b: Four-Sample Asymmetric Case for Power vs Sample Size with Sigma 5.0 .....	30
Figure 4.6.c: Four-Sample Asymmetric Case for Power vs Sample Size with Sigma 10.0 .....	30
Figure 4.6.d: Four-Sample Asymmetric Case for Power vs Sample Size with Sigma 20.0 .....	30

# LIST OF TABLES

Table 3.1: Summary of Parameter Settings for Simulations .....	15
--	----

# ACKNOWLEDGEMENTS

I would like to thank Dr. Leigh Murray, Dr. James Neill, and Dr. Weixing Song for contributing in the development of this report. I thank Dr. Murray for her role as major professor; for her guidance through retrospective power; and for her topical and gritty digressions on statistical consulting. I thank Dr. Neill for sitting on my committee; for always having a moment for questions; and for his ability to impart calm and fortitude during the rapidity and rigor of graduate studies. I thank Dr. Song for acting as a member of my committee; for helping me with SAS; and for his staunch encouragement and winning sense of humor.

I would also like to thank the faculty at Chapman University for preparing me for graduate studies. I owe much of my personal and academic growth to Dr. Atanas Radenski, Dr. Peter Jispen, Dr. Drew Moshier, Dr. Adrian Vajiac, and Dr. Mihaela Vajiac. For all the people I met at Chapman University, you are the most appreciated.

I would like to thank the friends I have made through the K-State Statistics department. I thank Seth Demel for all-night study sessions, Lee Goerl for 'talking stats' at social functions, and Aleksandra Banasik for discussing teaching strategies over early morning coffee. You made home feel larger than southern California.

Finally, I would like to thank my mother Valerie Chernoff, my sister Elyse Felder, and the friends I hold dear for putting up with all this 'Kansas/Graduate School' business. Your thoughts and affections are a constant source of inspiration.

## 1 Introduction

For many researchers, failing to reject the null hypothesis of statistical test is disconcerting. In some cases, the research design is questioned: was the study powerful enough to detect a meaningful effect? While valid statistical methods exist for calculating power before data have been collected, determining power retrospectively is seen in the literature as controversial. Here we explore power, as described in Thomas (1997), as a construct of sample data using the balanced one-way ANOVA model. Section 2 discusses methods for calculating prospective power; methods for estimating retrospective power; and concerns associated with estimating retrospective power. Section 3 outlines the balanced one-way ANOVA model; a method for estimating power using the observed effect, variance, sample size, and set significance level; and a method for constructing a minimal  $(1 - \alpha)$  100% confidence interval for true power. In addition, Section 3 contains conditions for simulations and implementation methods used for simulations. Section 4 summarizes the simulation results, displaying the location of true power, median estimated power, average estimated power, average p-value of tests and average confidence limits produced from the simulations. Section 4 also discusses the effect of population conditions on power calculations. Flow charts, code, and complete output are given in appendices A-D. By examining power after data have been collected and statistical tests have been performed, the perspective shifts from calculating power to estimating power.

## 2 Literature Review

Calculating the power of a statistical test before data have been collected, denoted *prospective power*, is generally regarded by statisticians as a good practice for obtaining conclusive results (American Statistical Association 1999). Power calculations are useful in determining adequate sample sizes for an up coming study (Lenth 2001). The power of a statistical test can be computed using a set significance level, sample size, effect, and variance (Lenth 2001; Gerard et al. 1998). Lenth (2001) provides a list of discussion-techniques for eliciting meaningful effect and variance values for a potential study. Power for t-tests, multiple regression, one-way anova, and other common statistical methods can be computed using the SAS procedure POWER (SAS Institute Inc. 2004b). Power for linear models with fixed class effects and contrast statements can be calculated using the SAS procedure GLMPOWER (SAS Institute Inc. 2004a). For many researchers, prospective power calculations are within reach.

Calculating the power of a statistical test after data have been collected, i.e. *retrospective power*, is less accepted by many statisticians as compared to prospective power (Hoenig and Heisey 2001; Thomas 1997; Yuan and Maxwell 2005). Retrospective power has been proposed for interpreting results when the null hypothesis cannot be rejected (Taylor and Muller 1995; Hogarty and Kromrey 2001) and for determining the sample size needed for the observed data to show statistically significant results (Steidl et al. 1997; Thomas 1997). Some popular methods for calculating retrospective power included: an observed sample size and given significance level where effect and variance come from literature (Yuan and Maxwell 2005); an observed effect, observed sample size, and given significance level (Yuan and Maxwell 2005) where variance comes from literature; an observed variance, observed sample size, and given significance level (Thomas 1997; Yuan and Maxwell 2005; Steidl et al. 1997) where effect comes from literature; and an observed effect, an observed sample size, an

observed variance, and given significance level (Gerard et al. 1998; Thomas 1997; Hoenig and Heisey 2001). Of these four methods only the last method is reviewed here.

Calculating retrospective power using an observed effect, an observed variance, an observed sample size, and given significance level has been shown to produce biased estimates when using an observed noncentrality parameter (Gerard et al. 1998). Thomas (1997) suggests adjusting the observed noncentrality parameter as shown in Wright and O'Brien (1988) and Johnson et al. (1995) to remove the bias in calculating power. The adjusted observed noncentrality parameter can produce negative values, which are typically set to zero (Gerard et al. 1998). A median estimator for the noncentrality parameter has been proposed by Taylor and Muller (1996) as a means for correcting the retrospective power calculation. The median estimator for the noncentrality parameter under-estimates half the time and over-estimates half the time. Here, the adjusted observed noncentrality parameter is explored.

Concerns about using retrospective power procedures to represent true power exist in the literature. Zumbo and Hubley (1998) provide a Bayesian argument against retrospective power as a logical representative for true power. Hoening and Heisey (2001) note that retrospective power obtain through observed values is redundant with the p-value of the test.

Due to the bias associated with retrospective power, confidence intervals for effect have been proposed in lieu of retrospective power analyses when the null hypothesis cannot be rejected (Gerard et al. 1998; Thomas 1997; Steidl et al. 1997).

In this paper, we examine the method presented in Thomas (1997) for estimating the power of a test in the context of the balanced one-way ANOVA model.

### 3 Methods

#### 3.1 Model

To investigate post-hoc power, the balanced one-way analysis of variance model is considered. The following briefly reviews this model and the usual analysis as described in Kuehl (2000) and Milliken and Johnson (2009). The means model for this situation is

$$y_{ij} = \mu_i + \varepsilon_{ij}, \text{ with } \varepsilon_{ij} \text{ 's i.i.d } N(0, \sigma^2) \text{ for } i = 1, \dots, p, j = 1, \dots, n. \quad (1.1)$$

Here  $y_{ij}$  represents the observed value of the  $j$ th observation from the  $i$ th treatment group and  $\mu_i$  denotes the population mean of the  $i$ th treatment group. The term  $\varepsilon_{ij}$  represents the random error associated with the  $j$ th observation from the  $i$ th treatment group. The errors are assumed independent and identically distributed from a normal population with mean 0 and variance  $\sigma^2$ .

The null and alternative hypotheses are that of no difference and that of some difference among the  $p$  population means, respectively:

$$\begin{aligned} H_0 : \mu_1 &= \dots = \mu_p \\ H_a : &\text{at least two } \mu_i \text{'s not equal} \end{aligned} \quad (1.2)$$

When the sample means are not “too different” from each other we deem the data consistent with the null hypothesis of equal means. We do not conclude that the population means are all the same but acknowledge that sample means are too similar to be called different. When at least two sample means exist which “strongly” differ we favor the alternative hypothesis and conclude that not all of the population means are equal to one another.

Two estimates of variance are used to measure the amount of evidence against the null hypothesis. The first estimate of variance is based only on the assumptions of the model. The second estimate of variance is based on the assumptions of the model and the assumption that



the null hypothesis is true. Of these two estimates of variance, the second estimate has a larger expected mean square than the first estimate when the null hypothesis is false.

The mean square for error is the first estimate of  $\sigma^2$  and is calculated by taking the average of the sample variances  $s_1^2, s_2^2, \dots, s_p^2$ :

$$\begin{aligned}
 MSE &= \frac{s_1^2 + s_2^2 + \dots + s_p^2}{p} \\
 &= \frac{1}{p(n-1)} \sum_{i=1}^p \sum_{j=1}^n (y_{ij} - \bar{y}_{i.})^2 \\
 &= \frac{1}{p(n-1)} \left\{ \sum_{i=1}^p \sum_{j=1}^n y_{ij}^2 - n \sum_{i=1}^p \bar{y}_{i.}^2 \right\} \\
 &= \frac{1}{p(n-1)} \left\{ \sum_{i=1}^p \sum_{j=1}^n y_{ij}^2 - n \bar{Y}^T \bar{Y} \right\} \\
 &= \frac{SSE}{DFE},
 \end{aligned} \tag{1.3}$$

where  $\bar{y}_{i.} = \frac{1}{n} \sum_{j=1}^n y_{ij}$ ,  $\bar{Y}^T = [\bar{y}_{1.}, \bar{y}_{2.}, \dots, \bar{y}_{p.}]$ , SSE is the error sum of squares, and DFE is the

error degrees of freedom. Under the assumptions of the model,

$$\frac{(DFE)(MSE)}{\sigma^2} \sim \chi_{(DFE)}^2 \tag{1.4}$$

and

$$E(MSE) = \sigma^2. \tag{1.5}$$

The mean square for treatments ( $MST$ ) is the second estimate of  $\sigma^2$  and is calculated as the sample variance of the sample means  $\bar{y}_{1.}, \bar{y}_{2.}, \dots, \bar{y}_{p.}$ . That is,

$$\begin{aligned}
MST &= \frac{n}{p-1} \sum_{i=1}^p (\bar{y}_{i.} - \bar{y}_{..})^2 \\
&= \frac{n}{p-1} \left\{ \sum_{i=1}^p \bar{y}_{i.}^2 - p \bar{y}_{..}^2 \right\} \\
&= \frac{1}{p-1} \left\{ \sum_{i=1}^p \frac{y_{i.}^2}{n} - \frac{y_{..}^2}{np} \right\} \\
&= \frac{1}{p-1} \left\{ n \bar{Y}^T \bar{Y} - \frac{n}{p} (\bar{Y}^T J)^2 \right\} \\
&= \frac{SST}{DFT},
\end{aligned} \tag{1.6}$$

where  $y_{i.} = \sum_{j=1}^n y_{ij}$ ,  $y_{..} = \sum_{i=1}^p \sum_{j=1}^n y_{ij}$ ,  $J^T = [1, 1, \dots, 1]$ , SST is the treatment sum of squares, and

DFT is the treatment degrees of freedom. Under the assumptions of the model and the assumption that the null hypothesis is true

$$\frac{(DFT)(MST)}{\sigma^2} \sim \chi_{(DFT)}^2 \tag{1.7}$$

and

$$E(MST) = \sigma^2. \tag{1.8}$$

Under the assumptions of the model and the assumption that the null hypothesis is not true

$$\frac{(DFT)(MST)}{\sigma^2} \sim \chi_{(DFT, \lambda)}^2 \tag{1.9}$$

and

$$E(MST) = \sigma^2 + \frac{n}{p-1} \sum_{i=1}^p (\mu_i - \bar{\mu}_{..})^2, \tag{1.10}$$

where

$$\lambda = \frac{n}{\sigma^2} \sum_{i=1}^p (\mu_i - \bar{\mu}_{..})^2 \tag{1.11}$$

denotes the noncentrality parameter.

Under the assumptions of the model and assumption that the null hypothesis is true,

$$\frac{\left[ (DFT)(MST) / \sigma^2 \right] / DFT}{\left[ (DFE)(MSE) / \sigma^2 \right] / DFE} \sim F_{(DFT, DFE)}, \quad (1.12)$$

where (1.7) and (1.9) are independently distributed  $\chi^2$  random variables. Under the assumptions of the model and the assumption that the alternative hypothesis is true,

$$\frac{\left[ (DFT)(MST) / \sigma^2 \right] / DFT}{\left[ (DFE)(MSE) / \sigma^2 \right] / DFE} \sim F_{(DFT, DFE, \lambda)}. \quad (1.13)$$

The observed  $F$ -statistic for the test is calculated as the ratio of the mean square for treatments over the mean square for error. This statistic is used to measure the amount of evidence in the sample data against the null hypothesis,

$$F = \frac{MST}{MSE} = \frac{\left[ (DFT)(MST) / \sigma^2 \right] / DFT}{\left[ (DFE)(MSE) / \sigma^2 \right] / DFE} \quad (1.14)$$

Given the assumptions of the means model and the assumption that the null hypothesis is true, the mean square for error should be of similar size to the mean square for treatments, practically speaking. However, given the null hypothesis is false, the mean square for error should be smaller than the mean square for treatments. An uncommonly large  $F$ -statistic leads to the conclusion that not all population means are equal. The  $p$ -value of the test gives the probability of obtaining an  $F$ -statistic greater than or equal to the one observed. The more evidence in the sample data against the null hypothesis of equal means the smaller the  $p$ -value.

### 3.2 Estimated Power

Power is estimated using the observed effect and the observed sampling variance. Let the cumulative distribution function of the noncentral  $F$ -distribution be denoted as

$$F(DFT, DFE, \lambda) . \quad (2.1)$$

The power of the  $F$ -test is defined as

$$\text{power} = 1 - F(F_{crit} \mid DFT, DFE, \lambda) , \quad (2.2)$$

where  $F_{crit}$  is the  $100 \cdot (1 - \alpha)$  percentile from a central  $F$ -distribution with numerator degrees of freedom  $DFT$ , denominator degrees of freedom  $DFE$ , level of significance  $\alpha$ , and noncentrality parameter  $\lambda$  (Thomas 1997). A positively-biased estimate for the power of an  $F$ -test is

$$\widehat{\text{power}} = 1 - F(F_{crit} \mid DFT, DFE, \hat{\lambda}) \quad (2.3)$$

where  $\hat{\lambda} = SST / MSE$  (Thomas 1997). The estimated noncentrality parameter can be adjusted to remove the upward bias. Therefore, an unbiased estimate of power (Wright and O'Brien 1988, and Johnson et al. 1995, cited in Thomas 1997) is

$$\widehat{\text{power}}_{adj} = 1 - F(F_{crit} \mid DFT, DFE, \hat{\lambda}_{adj}) \quad (2.4)$$

where  $\hat{\lambda}_{adj}$  is the adjusted estimated noncentrality parameter and is computed as

$$\hat{\lambda}_{adj} = [\hat{\lambda} \cdot (DFE - 2) / DFE] - DFT . \quad (2.5)$$

Note that for  $\hat{\lambda} \cdot (DFE - 2) / DFE$  less than  $DFT$ ,  $\hat{\lambda}_{adj}$  can be negative.

Upper and lower confidence limits for the true power are therefore

$$\widehat{\text{power}}_U = 1 - F\left(F_{crit} \mid DFT, DFE, \hat{\lambda}_U\right) \quad (2.6)$$

and

$$\widehat{\text{power}}_L = 1 - F\left(F_{crit} \mid DFT, DFE, \hat{\lambda}_L\right), \quad (2.7)$$

where  $\hat{\lambda}_U$  and  $\hat{\lambda}_L$  satisfy  $F\left(F_{obs} \mid DFT, DFE, \hat{\lambda}_U\right) = \alpha_U$  and  $F\left(F_{obs} \mid DFT, DFE, \hat{\lambda}_L\right) = 1 - \alpha_L$  in

which  $\alpha_U$  is the upper tail probability and  $\alpha_L$  lower tail probability (Thomas 1997). The upper and lower tail probabilities  $\alpha_U$  and  $\alpha_L$  are those which define the  $100 \cdot (1 - \alpha_U - \alpha_L)$  percent confidence interval for true power.

### 3.3 Confidence Interval Optimization

Confidence limits for true power were minimized to explore retrospective power under a “best” case scenario. Confidence limits set under  $\alpha_L = \alpha_U = \alpha / 2$  are not generally optimal (i.e. narrowest confidence interval) because of the skewed nature of the noncentral  $F$ -distribution. Setting upper and lower probabilities equal may under-represent, or over-represent, true power.

For a user-specified level of significance  $\alpha$ , we consider the  $100(1 - \alpha)\%$  confidence interval for true power, where

$$\alpha = \alpha_L + \alpha_U. \quad (3.1)$$

We denote the inverse cumulative distribution function of the noncentral  $F$ -distribution by

$$F^{-1}\left(\alpha_c, DFT, DFE, \hat{\lambda}_{adj}\right), \quad (3.2)$$

where  $\alpha_c$  is a lower tail probability,  $DFT$  the numerator degrees of freedom,  $DFE$  the denominator degrees of freedom, and  $\hat{\lambda}_{adj}$  the noncentrality parameter. We denote the quantile associated with  $\alpha_L$  as

$$F_L = F^{-1}(\alpha_L, DFT, DFE, \hat{\lambda}_{adj}), \quad (3.3)$$

and the quantile associated with  $\alpha_U$  as

$$F_U = F^{-1}(1 - \alpha_U, DFT, DFE, \hat{\lambda}_{adj}). \quad (3.4)$$

The width of the  $100(1 - \alpha)\%$  confidence interval for true power is minimized by finding the distance between the lower and upper quantiles such that

$$W = F_U - F_L \quad (3.5)$$

is minimized. Note, for fixed  $\alpha$  we have the constraint that  $\alpha = \alpha_L + \alpha_U$  (3.1) which gives

$$\alpha_U = \alpha - \alpha_L. \quad (3.6)$$

As such, we wish to find the value of  $\alpha_L$  which minimize the distance

$$W(\alpha_L) = F^{-1}(1 - \alpha + \alpha_L, DFT, DFE, \hat{\lambda}_{adj}) - F^{-1}(\alpha_L, DFT, DFE, \hat{\lambda}_{adj}), \quad (3.7)$$

for

$$\alpha_L \in (0, \alpha). \quad (3.8)$$

A method for finding  $\alpha_L$  such that  $W(x)$  is minimal is by the Golden Section Search (GSS) as described in Press et al. (2007). For a unimodal function  $W(x)$  defined for  $x \in [a, b]$ , the Golden Section Search method seeks to bracket the value  $k$  in  $[a, b]$  such that  $W(k) < W(x)$  for all other  $x \in [a, b]$ . The value  $k$  is considered found when brackets are

reached whose distance is within a user-specified tolerance level. The GSS procedure is named for its use of the mathematical constant

$$\varphi = \frac{-1 + \sqrt{5}}{2} \approx 0.618034 . \quad (3.9)$$

The ends points  $a$  and  $b$  are used to construct two new possible bracket points. We calculate the first new point, which we call  $c$ , by

$$c = a + \varphi(b - a) , \quad (3.10)$$

and the second new point, which we call  $d$ , by

$$d = a + \varphi^2(b - a) . \quad (3.11)$$

These new points are then evaluated with the function  $W(x)$  such that

$$u = W(c) \quad (3.12)$$

and

$$v = W(d) . \quad (3.13)$$

One of two cases may arise:

$$(1) \ u > v$$

$$(2) \ u \leq v$$

For the first case (Figure 3.1.a) the value  $k$  must be in  $[a, c]$  since the function  $W(x)$  is

unimodal. The interval  $[a, c]$  will then have width  $c - a = \varphi(b - a)$  (3.10). For the second case

(Figure 3.1.b) the value  $k$  must be in  $[d, b]$ , again since the function  $W(x)$  is unimodal. Using

the equation  $d = a + \varphi^2(b - a)$  (3.11) and the fact that  $\varphi^2 = 1 - \varphi$ , it can be shown that

$b - d = \varphi(b - a)$ . Therefore,

Figure 3.1.a: Update Bracket Interval for Minimand for  $v$  less than  $u$

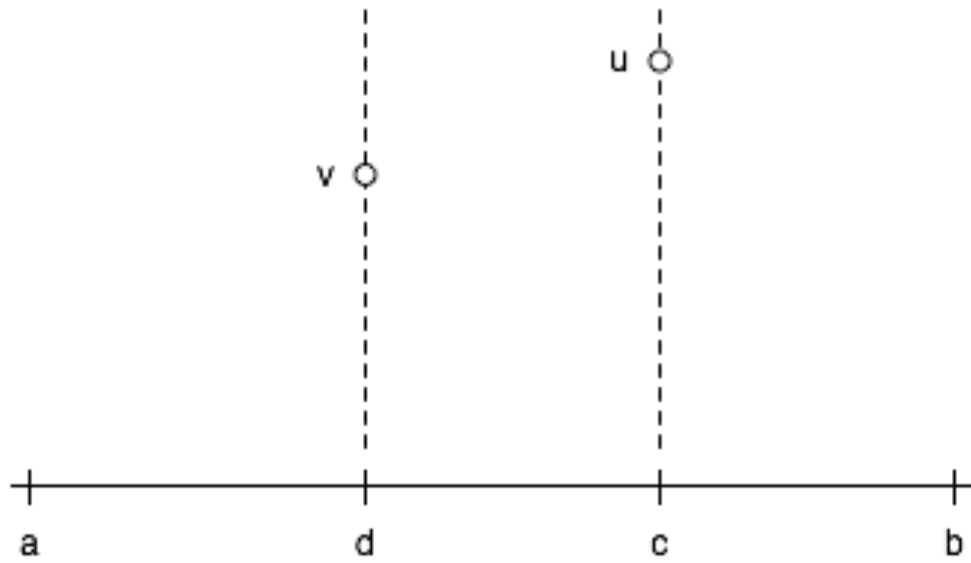
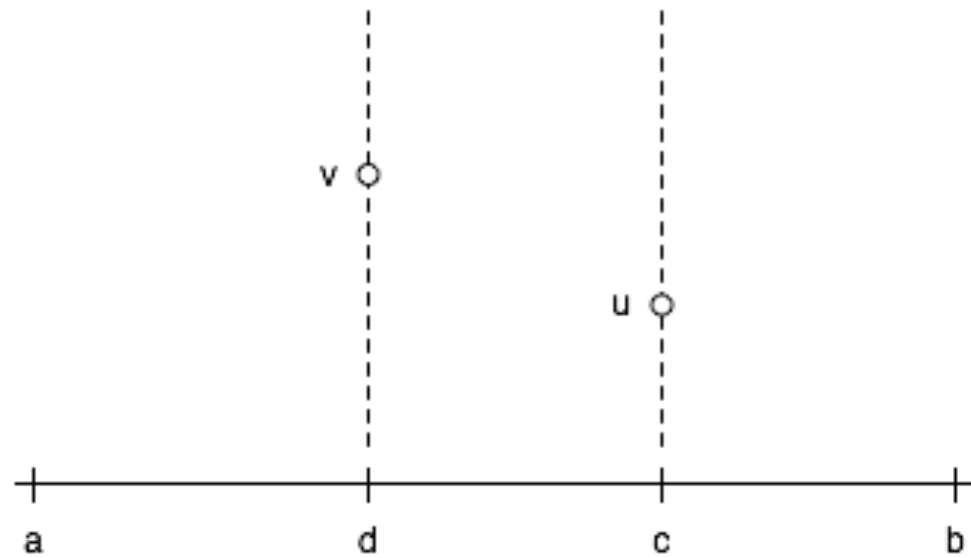


Figure 3.1.b: Update Bracket Interval for Minimand for  $v$  greater than  $u$





$$c - a = b - d , \quad (3.14)$$

that is, the interval width for case (1) equals the interval width for case (2). That is, we get a consistent reduction factor. Additionally,

$$\varphi = \frac{\varphi^2(b-a)}{\varphi(b-a)} = \frac{d-a}{b-d} \quad (3.15)$$

and

$$\varphi = -1 + \frac{1}{\varphi} = -1 + \frac{\varphi(b-a)}{\varphi^2(b-a)} = -\frac{d-a}{d-a} + \frac{\varphi(b-a)}{d-a} = \frac{-(d-a) + \varphi(b-a)}{d-a} = \frac{c-d}{d-a} \quad (3.16)$$

so that

$$\frac{c-d}{d-a} = \frac{d-a}{b-d} \quad (3.16)$$

that is, using  $\varphi$  to pick  $c$  and  $d$  maintains the spacing among points as the brackets are updated. Therefore, we only need to calculate one new point and make one new function evaluation per interval reduction.

### 3.4 Parameter Settings

Based on the balanced one-way ANOVA model, true power and estimated power were obtained for a range of population conditions and simulation outcomes, respectively, for three sizes of number of treatments,  $p = 2, 3$ , and 4.

The two-treatment class has a single symmetric mean arrangement. The three-treatment class has one symmetric mean arrangement and one asymmetric mean arrangement. The four-treatment class has two symmetric mean arrangements and one asymmetric mean arrangement. For each mean arrangement a range of population standard deviations and sample sizes are considered. These population conditions are summarized in Table 3.1. Table 3.1 also includes under each mean arrangement the sum of squared differences between population means and their grand population mean which is the numerator of the noncentrality parameter  $\lambda$  (1.10). These values were chosen so that treatment populations should be discernibly different at a 95% empirical interval when the population standard deviation is “small” and/or sample size is “large,” and indiscernibly different when the population standard deviation is “large” and/or sample size is “small.”

Table 3.1: Summary of Parameter Settings for Simulations

Case	Mean = $\mu_i$	$\sum(\mu_i - \bar{\mu}_\bullet)^2$
Two-Sample Symmetric	30, 40	50.0000
Three-Sample Symmetric	30, 35, 40	50.0000
Three-Sample Asymmetric	30, 40, 40	66.6667
Four-Sample Symmetric A	30, 35, 35, 40	50.0000
Four-Sample Symmetric B	30, 30, 40, 40	100.0000
Four-Sample Asymmetric	30, 40, 40, 40	75.0000
	Standard Deviation = $\sigma$	2.5 to 10.0 by 0.5 and 20.0
	Sample Size = $n$	5 to 14 by 1 and 15 to 50 by 5
	Significance Level = $\alpha$	0.05

### 3.5 Iteration

For each combination of mean arrangement, population standard deviation, and sample size, 100 simulations were produced. Each simulation produced  $p$  random samples where  $p$  denotes the number of treatment populations. The  $p$  samples were produced to have equal samples size  $n$ . Each random sample was produced from a random seed value, an integer between 1 and 1,000,000,000, obtained through the website [www.random.org](http://www.random.org). Random samples were constructed to reflect a normally distributed population with mean  $\mu_i$  and equal population standard deviation where  $i = 1, \dots, p$ . Note that random samples were initially generated in part by the SAS® software function RANNOR. RANNOR was seen to produce dependent samples when called multiple times within the IML procedure. Therefore, RANNOR was encapsulated in a macro which resulted in independent random number generation.

For each simulation, the following calculations were made from the generated data. An ANOVA table was calculated as outlined in 3.1.1. True power and estimated power were calculated as outlined in 3.1.2. Confidence limits for true power were calculated and optimized as outlined in 3.1.2 and 3.1.3, respectively. Tolerance for bracket intervals in 3.1.3 was set to 1E-4 as suggested by the procedure FMINBND in MATLAB (2009) and the package OPTIMIZE in R (R Development Core Team 2009).

Several problems were noted with the SAS software function PROBF which was recommended by Thomas (1997) for calculating power (SAS Institute Inc. 2006a). First, the SAS function PROBF reports an invalid argument when the supplied noncentrality parameter is large with respect to numerator degrees of freedom, denominator degrees of freedom, and  $F$  critical value. This problem is noted by O'Brien (1988). Taylor and Muller (1996) suggest setting the noncentrality large enough such that PROBF evaluates to zero or one where

appropriate. The SAS software function CDF (SAS Institute Inc. 2006b) offers some relief to the PROBF issue though invalid arguments can still be had for extreme parameter values.

Second, PROBF reports an error when the supplied noncentrality parameter is negative. Gerard et al. (1998) suggest setting the noncentrality parameter to zero. For some random seed values the data produced result with a negative adjusted estimated noncentrality parameter. The negative value occurs in some instances when adjusting the estimated noncentrality parameter (2.5), as noted previously.

Finally, FNONCT reports an error when either of the probabilities  $1 - \alpha_L$  or  $\alpha_U$  are greater than the probability associated with the observed  $F$  statistic from a central  $F$  - distribution with degrees of freedom DFT and DFE. Thomas (1997) suggests setting the noncentrality parameter equal to zero. In this case the error associated with FNONCT is noted in the SAS software documentation.

For each combination of mean arrangement, population standard deviation, and sample size, calculations for the 100 simulations are averaged. Notable averages include average estimated power and average upper and lower confidence limits about true power. The median for estimated power was also determined. Missing values were deleted.

The SAS program (Version 9.1.3) used can be found in Appendix B. Flow charts for the SAS program can be found in Appendix A (arranged through OmniGraffle Pro® version 5). This document was processed using the iWork '09® application Pages. Mathematical expression were set using MathType® version 6.

## 4 Results

Output for simulation summaries is plotted as power versus sample size for population standard deviations 2.5 to 10.0 by 0.5 and 20.0. Power ranges from 0.0 to 1.0 and sample size ranges from 5 to 14 by 1 and 15 to 50 by 5. The average p-value of the test is presented in each plot. The complete display of simulation plots can be found in Appendix C. In the Results, four plots are presented from each treatment class for each treatment mean arrangement, power plotted against sample size, where sample size ranges from 1 to 35 at population standard deviations of 2.5, 5.0, 10.0, and 20.0. A maximum sample size of 35 is presented since power approached 1.0 for sample sizes over 35 for the simulation configuration used here. Population standard deviations 2.5 and 20.0 represent two extreme cases for variability while population standard deviations 5.0 and 10.0 represent two medium cases for variability. Plots for each treatment arrangement appear as a moving-window like landscape from a moving vehicle.

#### 4.1 Two-Sample Symmetric Case

Plots for population standard deviations 2.5, 5.0, 10.0, and 20.0 with a maximum sample size of 35 are seen in Figure 4.1.a, 4.1.b, 4.1.c, and 4.1.d, respectively. The p-value for the test decreases as the sample size increases, such that the null hypothesis can be rejected when  $n \geq 5$ ,  $n \geq 6$ ,  $n \geq 14$ , and  $n > 35$  for respective values of  $\sigma = 2.5$ ,  $\sigma = 5.0$ ,  $\sigma = 10.0$ , and  $\sigma = 20.0$ . True power is always between the average lower and upper confidence limits for true power. Power and confidence limits are higher and closer together at larger sample sizes, as expected. For  $\sigma = 2.5$ , true power is very close to the average upper limit. For  $\sigma = 5.0$ , true power is closer to the average upper limit than the lower limit for  $n < 30$ ; is half-way between the average upper and lower limits when both limits and true power approach 1.0 for  $n \geq 30$ . For  $\sigma = 10.0$ , true power starts (at  $n = 5$ ) around the average lower limit, and is located half-way between average confidence limits at  $n = 9$ . For  $n > 9$ , true power is closer to the average upper limit than the average lower limit. When  $\sigma = 20.0$ , true power is located near the average lower limit (for  $n = 5$ ), and obtains an approximate location half-way between average limits for  $n = 30$ . True power exhibits an increasing positive trend toward the average upper limit for each plot, though the incline of the curve decreases for larger values of  $\sigma$ .

Power curves appear generally ordered as,

$$power_{true} > \widehat{power}_{med} > \widehat{power}_{avg} . \quad (4.1)$$

The general ordering of power curves is strictly seen for  $\sigma$  of 2.5 and 5.0. For  $\sigma = 10.0$ , average estimated power starts (at  $n = 5$ ) above median estimated power but adopts the general ordering for sample sizes 10 and greater. When  $\sigma$  equals 20.0, average estimated power starts (at  $n = 5$ ) above true power, and median estimated power starts below true power. Average estimated power appears (for  $\sigma = 20.0$ ) below true power at sample sizes 13 and greater, and below median estimated power at sample sizes 35 and greater. For samples sizes greater than 35, the general ordering appears to hold for any value of  $\sigma$  presented here.

Figure 4.1.a: Sigma 2.5

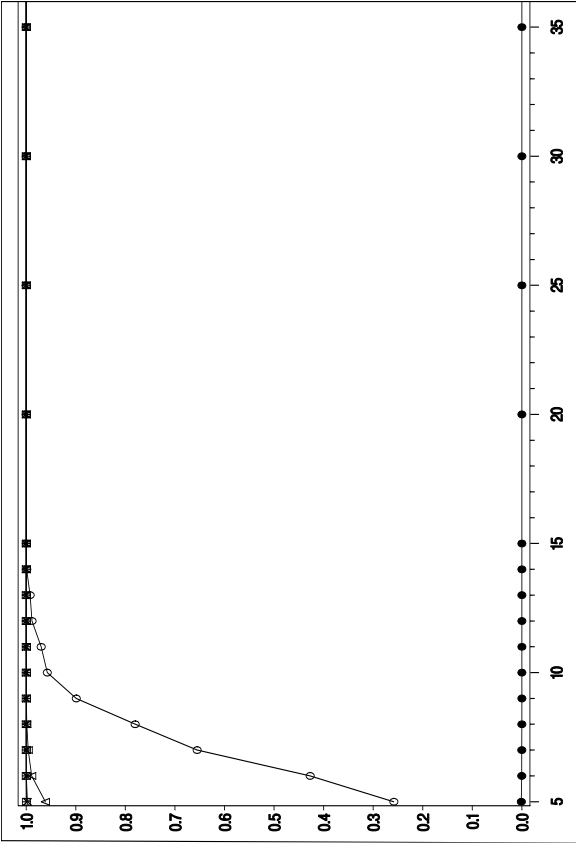


Figure 4.1.b: Sigma 5.0

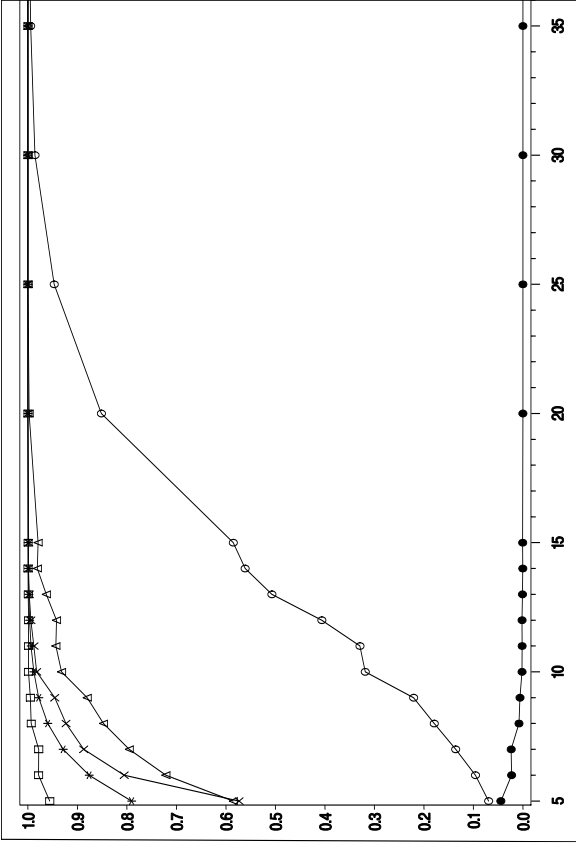


Figure 4.1.c: Sigma 10.0

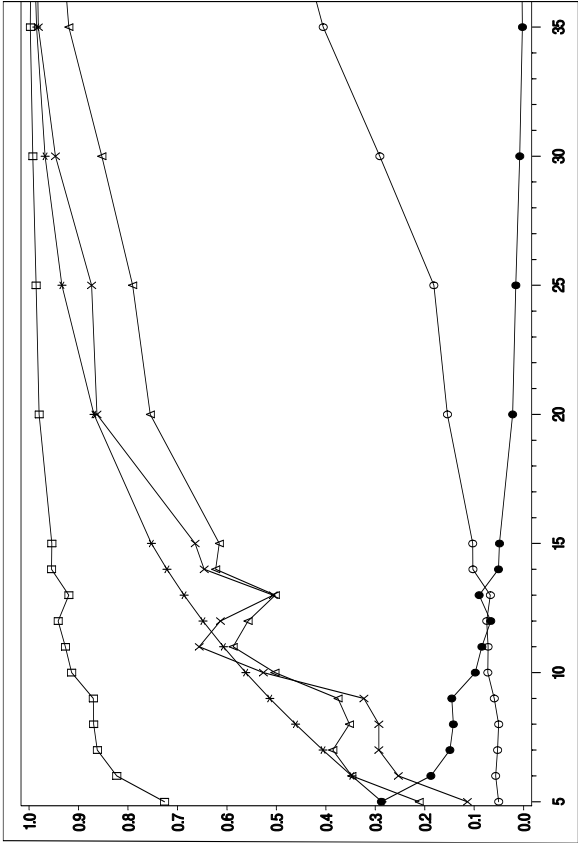
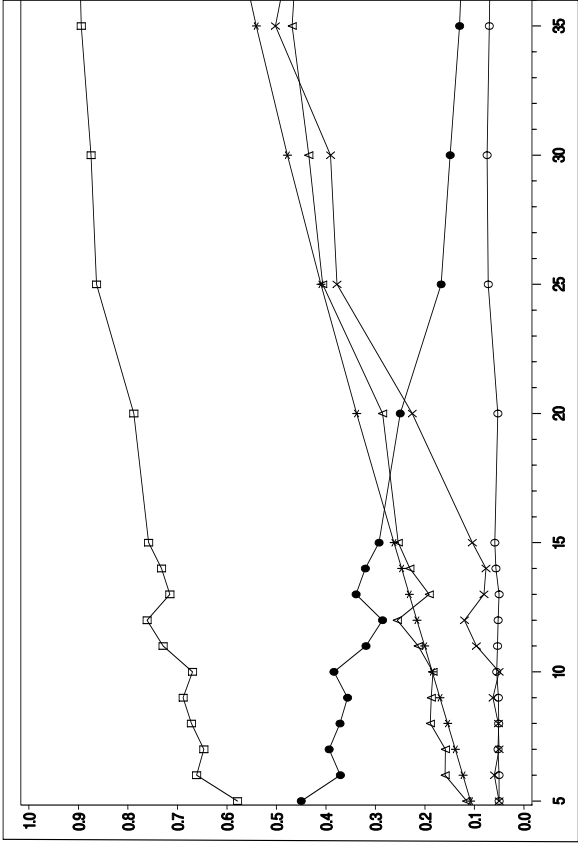


Figure 4.1.d: Sigma 20.0





## 4.2 Three-Sample Symmetric Case

Plots for population standard deviations 2.5, 5.0, 10.0, and 20.0 with a maximum sample size of 35 are seen in Figure 4.2.a, 4.2.b, 4.2.c, and 4.2.d, respectively. Similarly to the two-sample case, the p-value for the test decreases as the sample size increases, such that the null hypothesis can be rejected when  $n \geq 5$ ,  $n \geq 8$ ,  $n \geq 16$ , and  $n > 35$  for values of  $\sigma = 2.5$ ,  $\sigma = 5.0$ ,  $\sigma = 10.0$ , and  $\sigma = 20.0$ , respectively. True power is always contained within the average limits. As with the two-sample case, power and confidence limits tend to be higher and closer together for larger sample sizes. True power is very close to the average upper limit for  $\sigma = 2.5$ . For  $\sigma = 5.0$ , true power is located near the upper average limit and is approximately half-way between the average limits for  $n \geq 35$ . When  $\sigma = 10.0$ , true power is close to the average lower limit (at  $n = 5$ ), and is approximately half-way between the confidence limits at  $n = 12$ . For  $\sigma = 20.0$ , true power is nearer to the average lower limit than the average upper limit for sample sizes up to 30, but for  $n = 35$  true power is approximately half-way between the confidence limits. At all values  $\sigma$  considered here, true power tend toward the average upper limit.

As with the two-sample case, the general ordering (4.1) tends to hold. The general ordering of power is seen for  $\sigma = 2.5$  and 5.0. For  $\sigma = 10.0$ , average estimated power starts (at  $n = 5$ ) above the median estimated power, but takes on the general ordering at sample sizes 13 and greater. For  $\sigma = 20.0$ , true power begins (at  $n = 5$ ) below average estimated power, and above median estimated power. Average estimated power (for  $\sigma = 20.0$ ) is below true power at sample sizes 30 and greater, and below median estimated power at sample sizes greater than 35. The general position of power curves (4.1) tends to hold when samples sizes are large.

Figure 4.2.a: Sigma 2.5

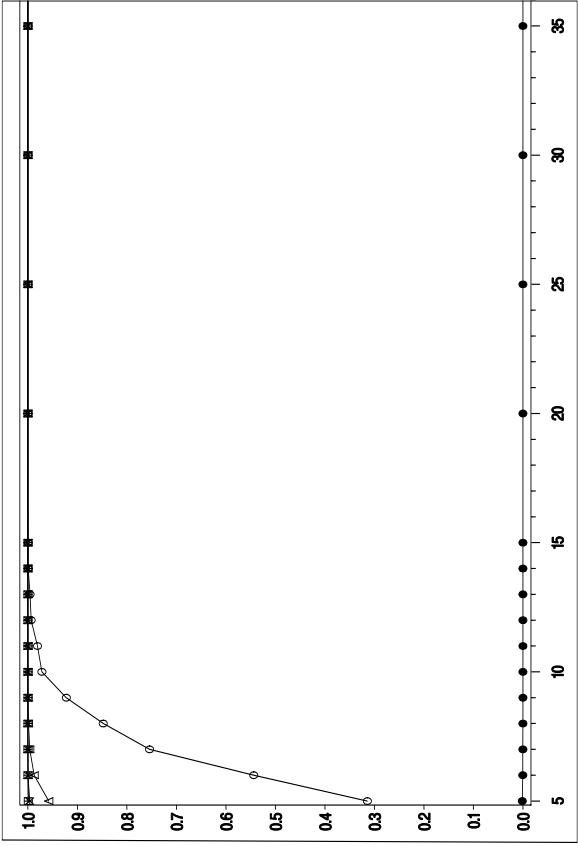


Figure 4.2.b: Sigma 5.0

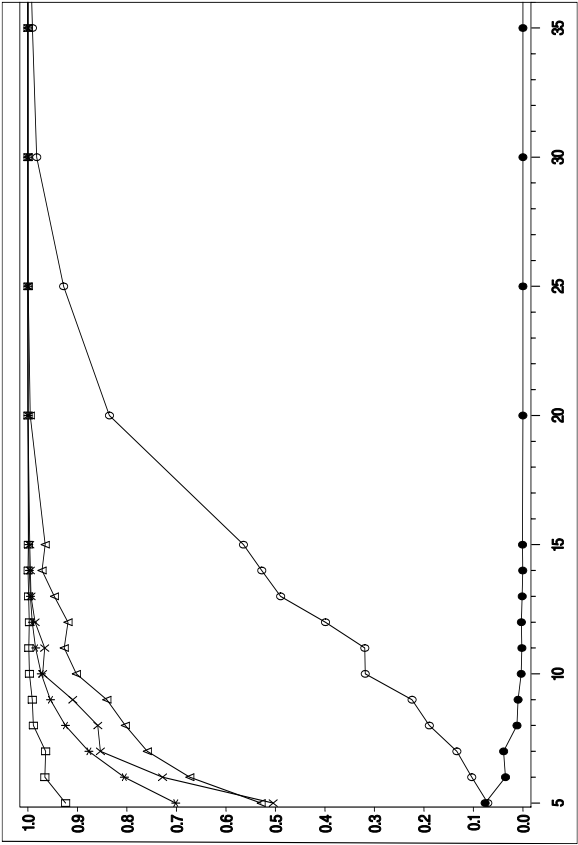


Figure 4.2.c: Sigma 10.0

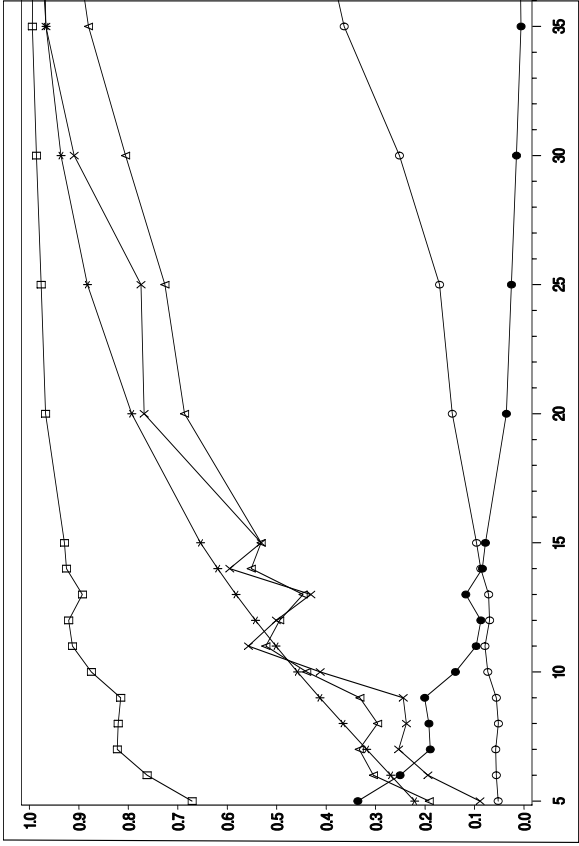
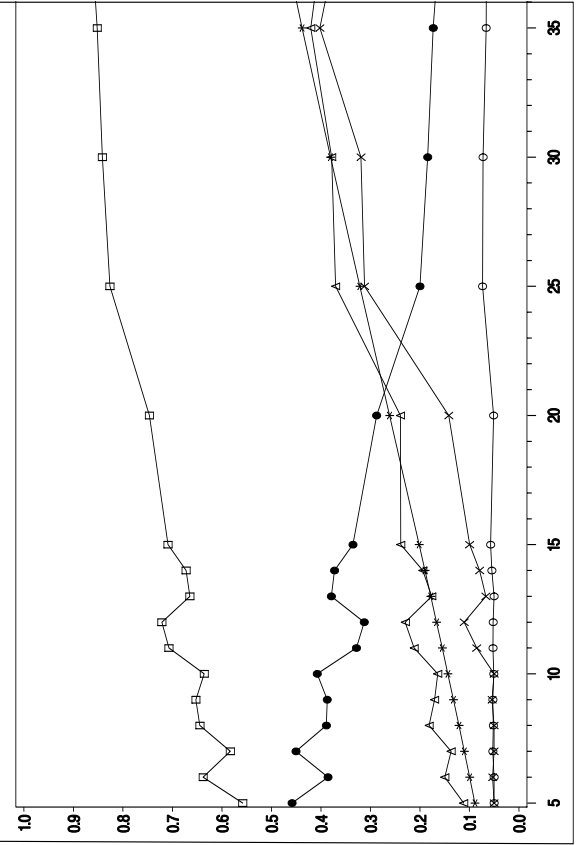


Figure 4.2.d: Sigma 20.0



### 4.3 Three-Sample Asymmetric Case

Plots for population standard deviations 2.5, 5.0, 10.0, and 20.0 with a maximum sample size of 35 are seen in Figure 4.3.a, 4.3.b, 4.3.c, and 4.3.d, respectively. Again, the p-value for the test decreases as the sample size increases, such that the null hypothesis can be rejected when  $n \geq 5$ ,  $n \geq 6$ ,  $n \geq 14$ , and  $n > 35$  for respective values of  $\sigma = 2.5$ ,  $\sigma = 5.0$ ,  $\sigma = 10.0$ , and  $\sigma = 20.0$ . True power is between the average upper and lower limits for all values of the population standard deviation considered here. As with the previous cases, power and confidence limits are higher and closer together for larger sample sizes. For  $\sigma = 2.5$ , true power is indistinguishable from the average upper limit. For  $\sigma = 5.0$ , true power is closer to the average upper limit than the average lower limit, and true power is approximately half-way between the average upper and lower limits when both limits approach 1.0 ( $n \geq 30$ ). For  $\sigma = 10.0$ , the average upper and lower limits are centered around true power for  $n = 9$ . When  $\sigma = 20.0$ , true power is located near the average lower limit (for  $n = 5$ ), and is approximately half-way between the average upper and lower limits at  $n = 30$ . As with previous cases, true power shows an upward trend toward the average upper limit for larger sample sizes.

As with prior cases, the general ordering of true power, median estimated power, and average estimated power (4.1) tends to hold. For  $\sigma = 2.5$  and 5.0, the general ordering of power curves can be seen in Figure 4.3.a. For  $\sigma = 10.0$ , average estimated power starts (at  $n = 5$ ) above the median estimated power, but adopts the general ordering for sample sizes larger than 10. When  $\sigma = 20.0$ , average estimated power starts (at  $n = 5$ ) above true power, and median estimated power starts (at  $n = 5$ ) below true power. Average estimated power appears below true power for sample sizes 15 and greater, and below median estimated power for sample sizes greater than 35. As with the previous cases, the general order of the power curves (4.1) is maintained for large sample sizes.

Figure 4.3.a: Sigma 2.5

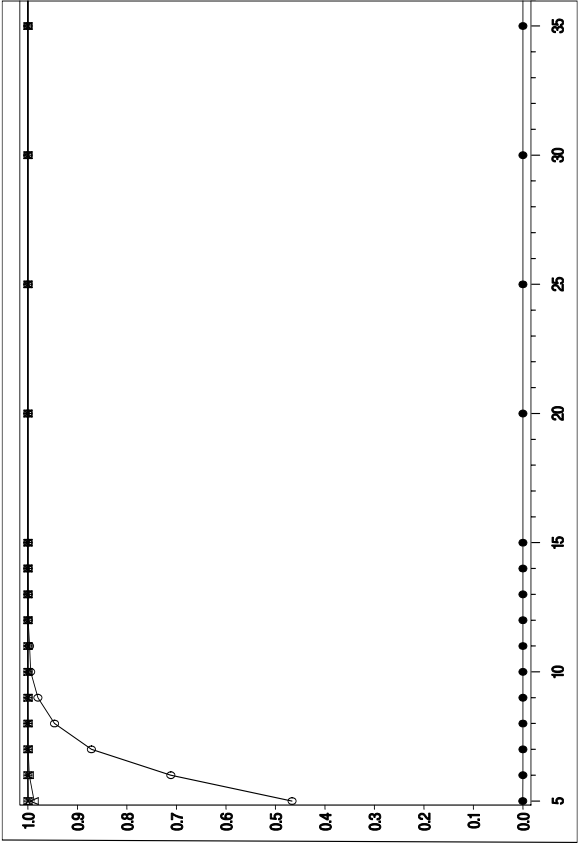


Figure 4.3.b: Sigma 5.0

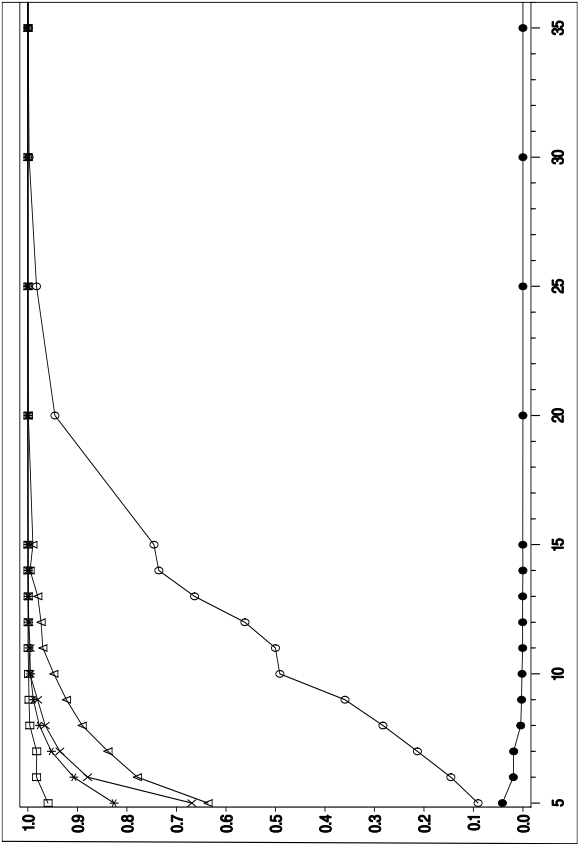


Figure 4.3.c: Sigma 10.0

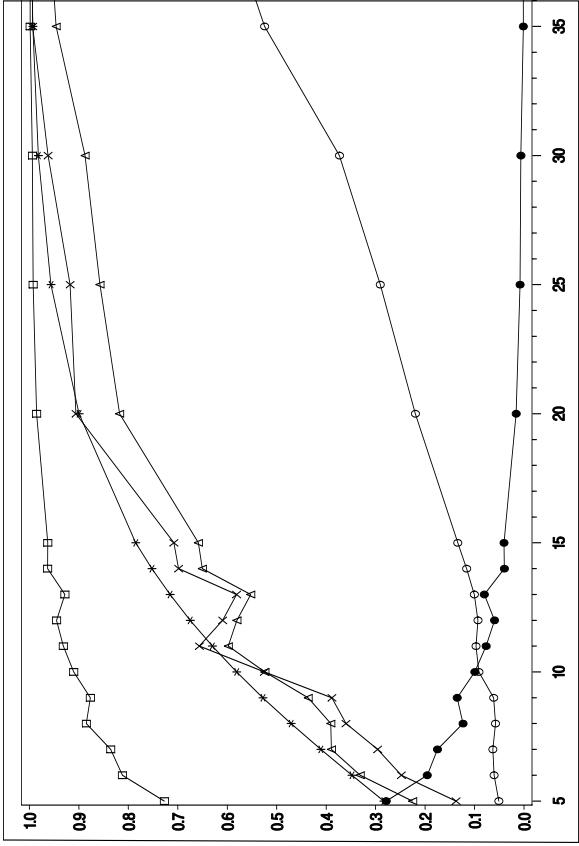
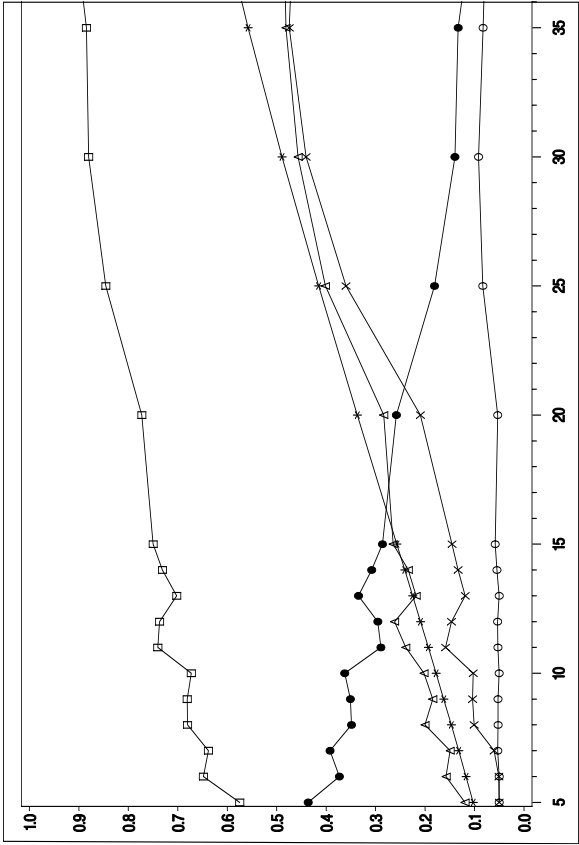


Figure 4.3.d: Sigma 20.0



#### 4.4 Four-Sample Symmetric Case A

Plots for population standard deviations 2.5, 5.0, 10.0, and 20.0 with a maximum sample size of 35 are seen in Figure 4.4.a, 4.4.b, 4.4.c, and 4.4.d, respectively. For larger sample sizes, the p-value for the test decreases such that the null hypothesis can be rejected when  $n \geq 5$ ,  $n \geq 8$ ,  $n \geq 17$ , and  $n > 35$  for respective values of  $\sigma = 2.5$ ,  $\sigma = 5.0$ ,  $\sigma = 10.0$ , and  $\sigma = 20.0$ . True power, as seen in previous cases, is always between the average upper and lower limits for true power. Further, true power and average confidence limits tend toward 1.0 for large sample sizes. For  $\sigma = 2.5$ , true power is identical with the average upper confidence limit. For  $\sigma = 5.0$ , average confidence limits center are symmetric about true power when both limits equal 1.0. For  $\sigma = 10.0$ , average upper and lower confidence limits are centered around true power when  $n = 14$ . For  $\sigma = 20.0$ , true power is close to the average lower limit. True power displays a positive increasing trend toward the average upper limit, where the steepness of the curve reduces for larger values of  $\sigma$ .

The general order of the power curves (4.1) again tends to occur. The typical ordering always occurs for  $\sigma = 2.5$  and 5.0. For  $\sigma = 10.0$ , median estimated power starts (at  $n = 5$ ) below the median estimated power, but maintains the typical ordering at sample sizes 14 and greater. For  $\sigma = 20.0$  and  $n = 5$ , power curves are ordered as average estimated power, true power, and median estimated power. The average estimated power (for  $\sigma = 20.0$ ) appears below true power at sample sizes 15 and greater, and below median estimated power at sample sizes larger than 35. As with previous cases, the typical order of power curves (4.1) is seen for larger sample sizes.

Figure 4.4.a: Sigma 2.5

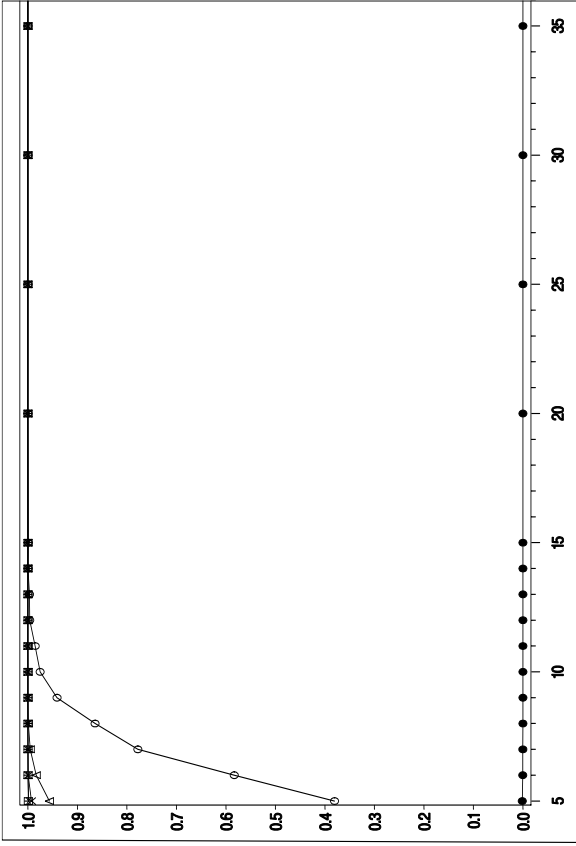


Figure 4.4.b: Sigma 5.0

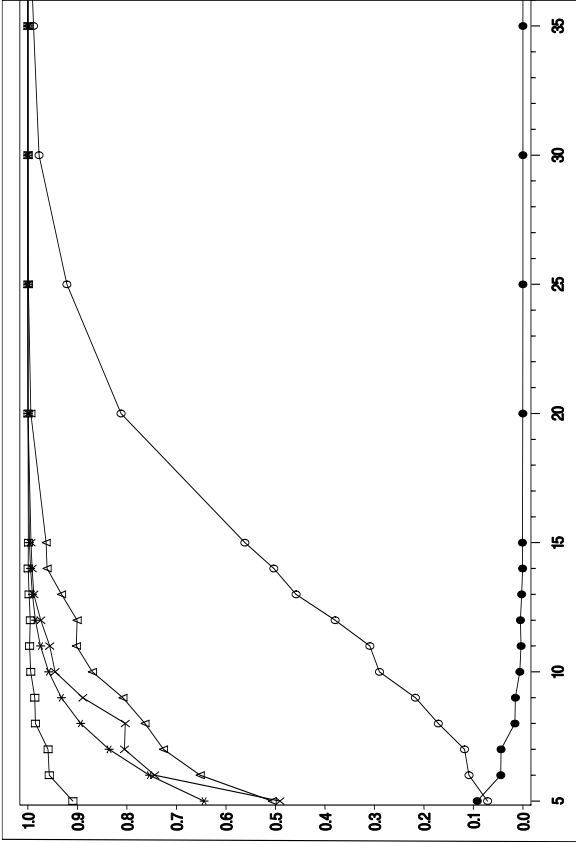


Figure 4.4.c: Sigma 10.0

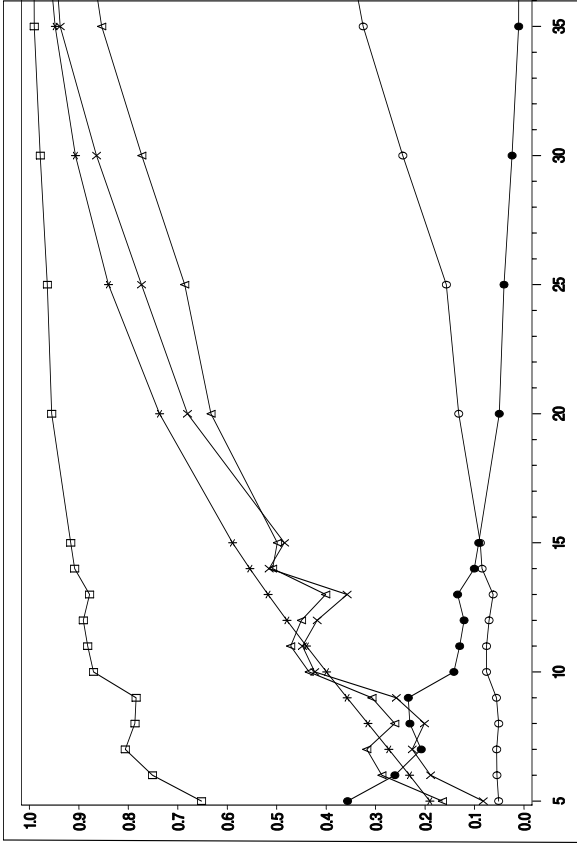
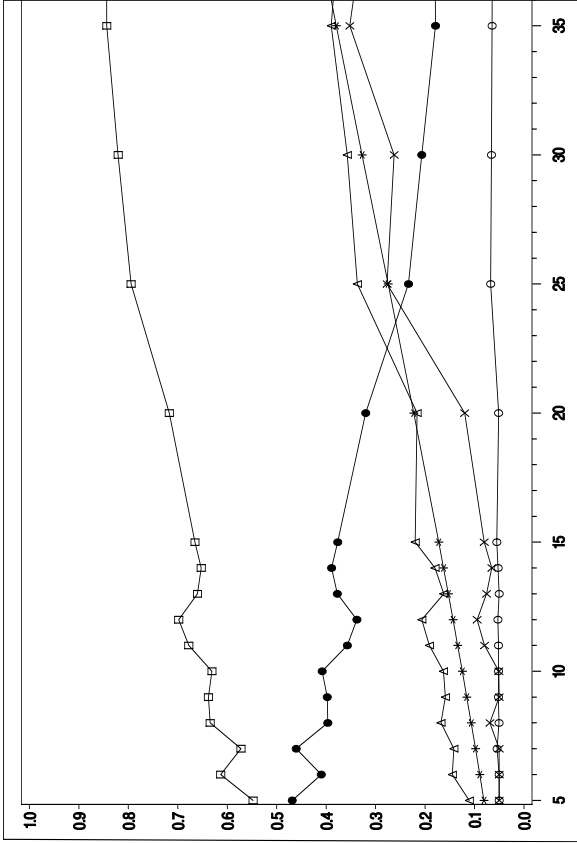


Figure 4.4.d: Sigma 20.0



#### 4.5 Four-Sample Symmetric Case B

Plots for population standard deviations 2.5, 5.0, 10.0, and 20.0 with a maximum sample size of 35 are seen in Figure 4.5.a, 4.5.b, 4.5.c, and 4.5.d, respectively. As the sample size increases, the p-value for the test decreases such that the null hypothesis can be rejected when  $n \geq 5$ ,  $n \geq 5$ ,  $n \geq 10$ , and  $n > 35$  for  $\sigma = 2.5$ ,  $\sigma = 5.0$ ,  $\sigma = 10.0$ , and  $\sigma = 20.0$ , respectively. As with previous cases, average lower and upper confidence limits always contain true power. True power and confidence limits become narrower and approach 1.0 for larger sample sizes. For  $\sigma = 2.5$ , true power is identical to the average upper limit. For  $\sigma = 5.0$ , true power is just below the average upper limit and is centered between average limits when both limits approach 1.0. For  $\sigma = 10.0$ , true power begins (at  $n = 5$ ) around the average lower limit, and centers between the average confidence limits for  $n = 8$ . When  $\sigma = 20.0$ , average confidence limits are symmetric around true power for  $n = 25$ . As with prior cases, true power has an increasing positive trend toward the average upper limit for each plot. Notably, the rate of increase of the true power curve decreases for larger values of  $\sigma$ .

As seen previously, average estimated power, median estimated power, and true power are consistently between the lower and upper confidence limits for true power. Power curves have the typical order (4.1) as in previous cases. The usual ordering of power is seen for  $\sigma = 2.5$  and 5.0. For  $\sigma = 10.0$ , average estimated power starts (at  $n = 5$ ) above the median estimated power, yet takes on the general ordering at sample sizes 10 and greater. For  $\sigma = 20.0$ , true power is below average estimated power and above median estimated power for  $n = 5$ . Further, for  $\sigma = 20.0$ , average estimated power appears below true power at sample sizes 13 and greater, and below median estimated power at sample sizes 30 and greater. As seen earlier, the general ordering (4.1) holds for large sample sizes.

Figure 4.5.a: Sigma 2.5

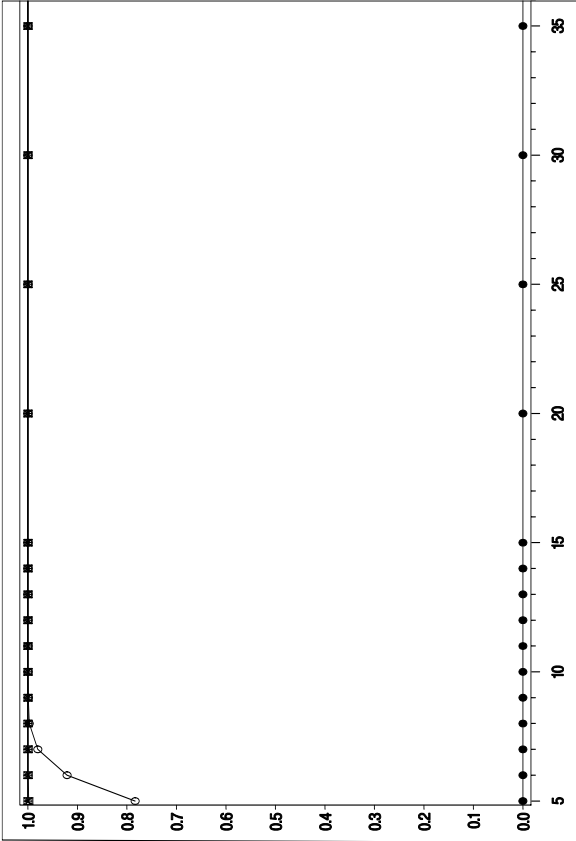


Figure 4.5.b: Sigma 5.0

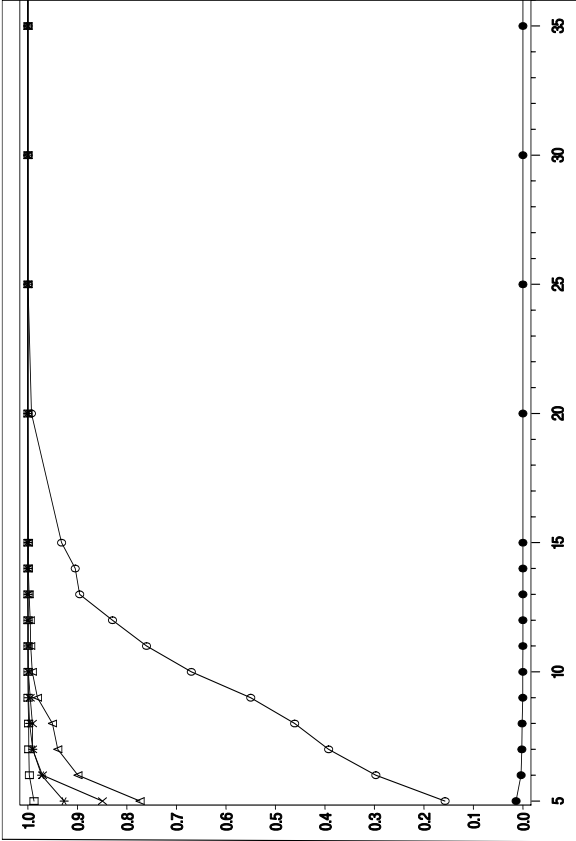


Figure 4.5.c: Sigma 10.0

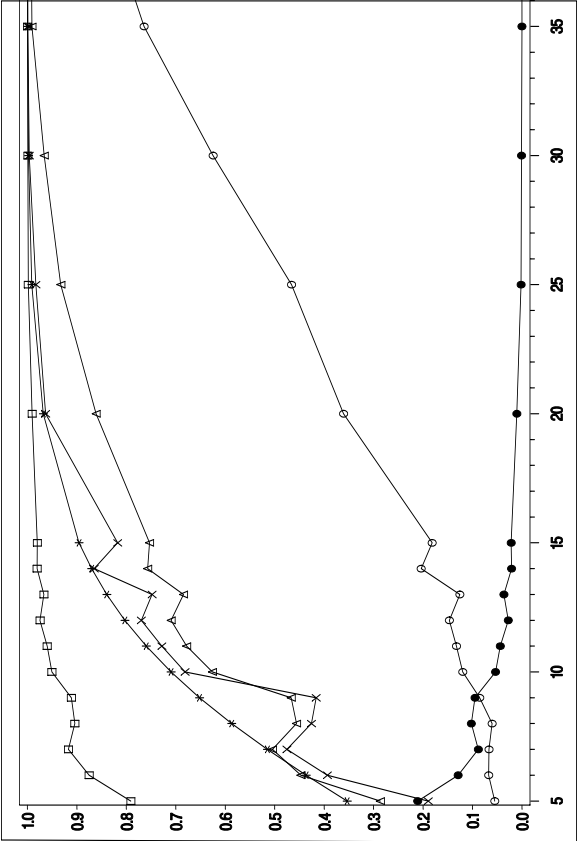
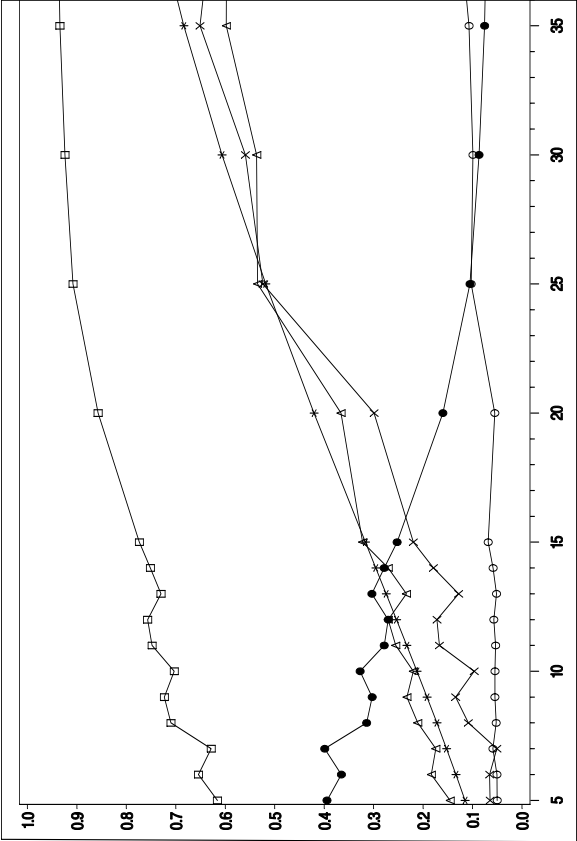


Figure 4.5.d: Sigma 20.0





#### 4.6 Four-Sample Asymmetric Case

Plots for population standard deviations 2.5, 5.0, 10.0, and 20.0 with a maximum sample size of 35 are seen in Figure 4.6.a, 4.6.b, 4.6.c, and 4.6.d, respectively. The p-value for the test decreases as the sample size increases, as expected, such that the null hypothesis can be rejected when  $n \geq 5$ ,  $n \geq 6$ ,  $n \geq 14$ , and  $n > 35$  for respective values of  $\sigma = 2.5$ ,  $\sigma = 5.0$ ,  $\sigma = 10.0$ , and  $\sigma = 20.0$ . True power lies between the average lower and upper limits for all values of  $\sigma$ . As seen in prior cases, true power and average confidence limits are higher and closer together for large sample sizes. For  $\sigma = 2.5$ , true power is indistinct from the average upper limit. For  $\sigma = 5.0$ , true power is close to the average upper limit and is approximately very close (at power approximately 1.0) with the average upper limit at  $n = 10$ . For  $\sigma = 10.0$ , average confidence limits are equally spaced about true power when  $n = 9$ . When  $\sigma = 20.0$ , average upper and lower limits are centered around true power for  $n = 30$ . As usual, true power shows an increasing upward trend toward the average upper confidence limit for each plot. Notably, the incline of the curve for true power climbs with less rapidity for larger values of  $\sigma$ .

The power curves are hold the usual ordering (4.1) as seen previously. The general ranking of power curves (4.1) is exemplified for  $\sigma = 2.5$  and 5.0. For  $\sigma = 10.0$ , average estimated power starts (at  $n = 5$ ) above the median estimated power but takes on the general ordering at sample sizes 9 and greater. For  $\sigma = 20.0$ , average estimated power starts above true power, while median estimated power starts below true power. Average estimated power (for  $\sigma = 20.0$ ) appears below true power at sample sizes 20 and greater, and below median estimated power at sample sizes greater than 35. When sample sizes are large, the power curves follows the general ordering (4.1).

Figure 4.6.a: Sigma 2.5

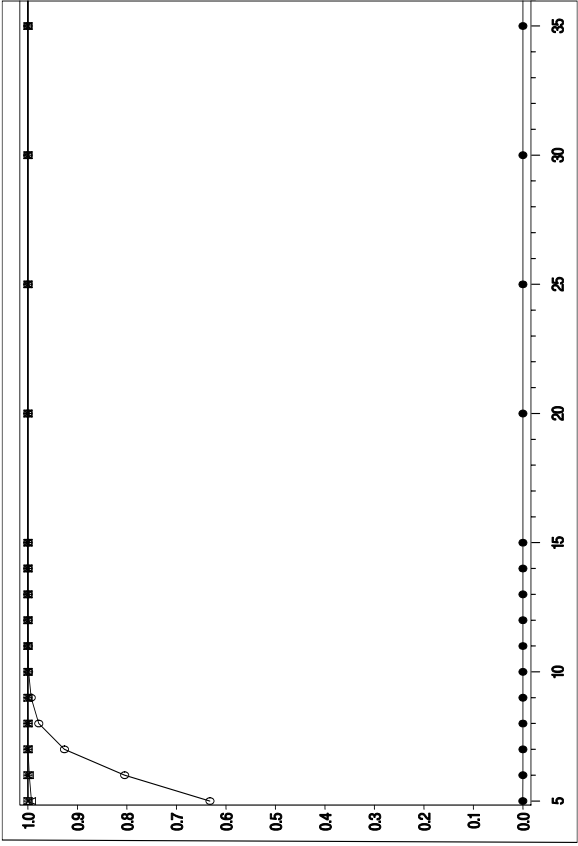


Figure 4.6.b: Sigma 5.0

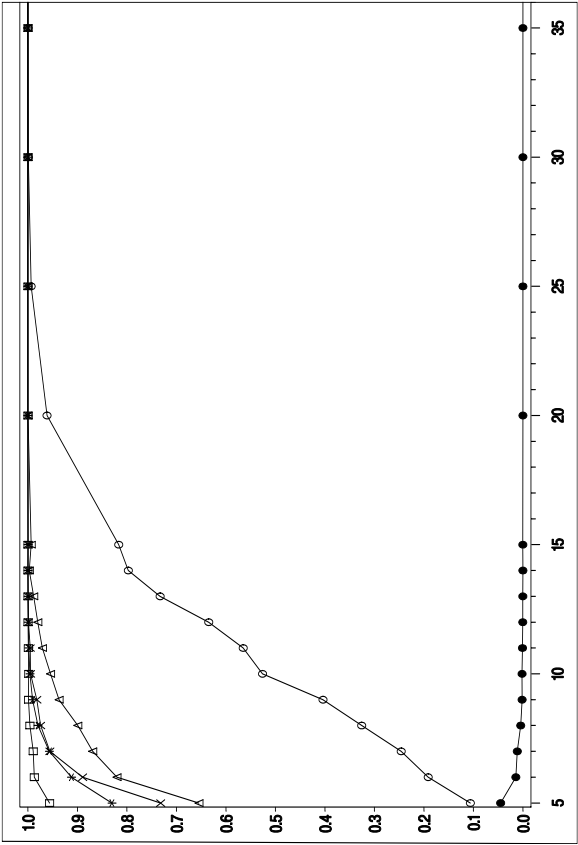


Figure 4.6.c: Sigma 10.0

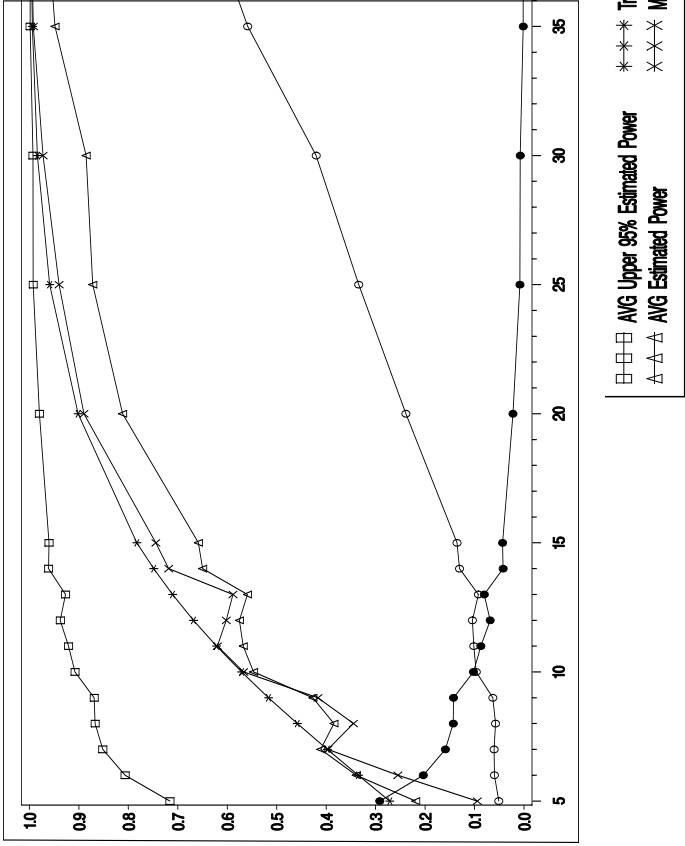
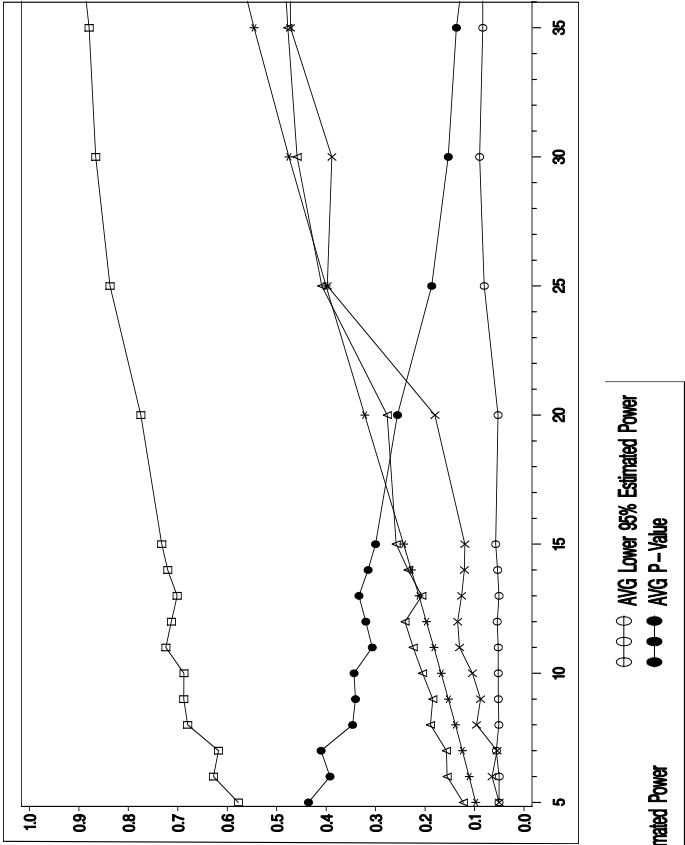


Figure 4.6.d: Sigma 20.0



#### 4.7 Symmetric Cases

For symmetric treatment arrangements, confidence limits for the two-sample case, three-sample case, and four-sample case A do not appear substantially different. As an exception, confidence limits for the four-sample case B are narrower than the other symmetric cases. Average lower and upper confidence limit widths are ordered, from least to greatest, as four-sample case B, four-sample case A, three-sample case, and two-sample case. Average lower and upper confidence limits always contain true power.

For all symmetric cases, true power shows a positive increasing trend toward the average upper limit. For  $\sigma = 2.5$  and  $5.0$ , all symmetric cases show true power near the average upper limit. For  $\sigma = 10.0$ , true power starts (at  $n = 5$ ) close to the average lower limit, reaches a half-way distance between average limits for  $n = 5$  to  $15$ , and then proceeds toward the average upper limit. For  $\sigma = 20.0$ , true power is generally close to the average lower limit for all symmetric cases, though the two-sample case and four-sample case B reach a half-way distance between average limits at  $n = 30$  and  $n = 25$ , respectively.

#### 4.8 Asymmetric Cases

For the two asymmetric cases, confidence interval widths are noticeably different from one another. The four-sample case has narrower average lower and upper confidence limits for true power than the three-sample case. True power is always between the average lower and upper confidence limits.

For both asymmetric cases, true power exhibits an upward trend toward the average upper limit. For  $\sigma = 2.5$  and  $5.0$ , true power is never half-way between the average upper and lower limits. For  $\sigma = 10.0$ , true power starts (at  $n = 5$ ) around the average lower limit, reaches the half-way distance between average limits around  $n = 8$  to  $9$ , and then moves closer to the

average upper limit. For  $\sigma = 20.0$ , true power is close to the average lower limit, but reaches the half-way distance between average limits around  $n = 30$  to  $35$ .

#### **4.9 Symmetric Cases versus Asymmetric Cases**

In both symmetric and asymmetric cases, true power is always between the average lower and upper limits. Generally, symmetric cases appear to have wider average confidence limits than the asymmetric cases, with the exception of the four-sample case B, which has the narrowest confidence limits among all cases considered. Symmetric and asymmetric cases both show that true power is closer to the average upper limit than the lower limit for large sample sizes. In terms of average limit widths, cases are ordered from least to greatest as follows: four-sample case B, four-sample asymmetric case, three-sample asymmetric case, four-sample case A, three-sample symmetric case, then the two-sample case.

Both symmetric cases and asymmetric cases have power curves typically ordered, from highest to lowest, as true power, median estimated power, then average estimated power. The typically ordering of power curves is seen for  $\sigma = 2.5$  and  $5.0$ . For  $\sigma = 10.0$ , average estimated power starts (at  $n = 5$ ) above median estimated power, but crosses below median estimated power for samples sizes between 10 to 15. For  $\sigma = 20.0$ , average estimated power is above true power, and median estimated power is below true power. Further (for  $\sigma = 20.0$ ), average estimated power drops below true power, then proceeds below median estimated power as sample sizes increase. All power curves are within the average lower and upper limits.

## 5 Conclusion

Simulations were constructed for a range of population conditions under the balanced one-way ANOVA model. Each simulation case was iterated 100 times and plots for average estimated power, median estimated power, true power, average lower 95% limit, and average upper 95% limit were produced. An average p-value for  $F$ -tests was also plotted for each simulation setup.

When the population standard deviation was small (from  $\sigma = 2.5$  to  $5.0$ ), confidence limits for true power were wider for small sample sizes and narrower for medium-to-large sample sizes. Under all sample sizes considered here, average limits contained true power. Notably, true power was never half-way between the average upper and lower confidence limits except when sample sizes were large and both limits were very close to 1.0. Additionally, true power was always close to the average upper limit and above the average estimate for power. When significant differences exist among the population treatment means, the average p-value is small, true power is large, and the average retrospective estimate for power provides a higher lower limit for true power than the average lower limit.

When the population standard deviation was large (from  $\sigma = 10.0$  to  $20.0$ ), average confidence limits for true power were generally wide for small and large sample sizes. True power was contained within the average limits: close to the average lower limit for small sample sizes and progressive toward the average upper limit for large sample sizes. In all cases, the average estimate for power was present above true power (when sample sizes were small) and below true power (when sample sizes were large). When the null hypothesis of equal population means was difficult to reject, i.e. when the p-value of the test was large, true power was low and confidence limits for true power were wide.

When the null hypothesis cannot be rejected, the test performed is seen as under powered. The results here show that true power is hard to determine retrospectively as laid out

in Thomas (1997) when the null hypothesis cannot be rejected, since the average 95% confidence limits were wide and the average estimate for power fails to act as a lower bound for true power under all values of  $\sigma$ .

The presence of nonsignificant results can lead researchers to question the power of their test. When power has not been calculated *a priori*, researchers may see *retrospective* methods as a viable technique for “figuring out what power was.” As an example of Thomas’ technique to calculate power *post hoc*, two possible consulting situations are described below. Each uses an observed  $F$ -statistic, numerator and denominator degrees of freedom, and the desired level of significance. The p-value of the test is also provided.

For the first example, consider an observed  $F$ -statistic of 1.5 with 1 numerator degree of freedom and 40 denominator degrees of freedom with a p-value of 0.2278. An estimate for true power is

$$\widehat{\text{power}} = 1 - F(4.085 | 1, 40, 0.425) = 0.0975 ,$$

where  $F_{0.05, 1, 40} = 4.085$  is the upper tail critical value from a central  $F$ -distribution with 1

numerator degree of freedom and 40 denominator degrees of freedom, and

$\hat{\lambda}_{adj} = ((1.5)(1)(40 - 2)) / 40 - 1 = 0.425$  is the adjusted estimated noncentrality parameter. The

minimized 95% confidence limits for true power are

$$\widehat{\text{power}}_L = 1 - F(4.085 | 1, 40, 0.00000) = 0.05 \text{ and}$$

$$\widehat{\text{power}}_U = 1 - F(4.085 | 1, 40, 8.2801704) = 0.8017 ,$$

where the estimated lower and upper noncentrality parameters  $\hat{\lambda}_L = 0.000000$  and

$\hat{\lambda}_U = 8.2801704$  are determined by the lower and upper tail probabilities  $\alpha_L = 0.000048$  and

$\alpha_U = 0.049952$  through use of the Golden Section Search method (Section 3.3). Using the estimate for power, a researcher would conclude that the test was under powered since this estimated value is below 0.8 (Cohen 1988). Notably, using the point estimate to represent true power is uninformative since the confidence limits have true power anywhere from 0.05 to 0.8017. The confidence interval here is similar to the confidence limits in Figure 4.1.d for the symmetric two population mean arrangement with  $\sigma = 20.0$  at a sample size of 22. Figure 4.1.d likens the clients' observations to an extreme case where discerning treatment groups as different is difficult.

For the second example, consider an observed  $F$ -statistic of 4.05 with 1 numerator degree of freedom and 40 denominator degrees of freedom with a p-value of 0.0509. Power is estimated as

$$\widehat{\text{power}} = 1 - F(4.085 | 1, 40, 2.8475) = 0.3773,$$

where  $F_{0.05, 1, 40} = 4.085$  is the upper tail critical value from a central  $F$ -distribution with 1

numerator degree of freedom and 40 denominator degrees of freedom, and

$\hat{\lambda}_{adj} = ((4.05)(1)(40 - 2)) / 40 - 1 = 2.8475$  is the adjusted estimated noncentrality parameter.

Through the Golden Section Search (Section 3.3), the minimized lower and upper tail probabilities are found to be  $\alpha_L = 0.000048$  and  $\alpha_U = 0.049952$  which yield lower and upper

estimated noncentrality parameters  $\hat{\lambda}_L = 0.000000$  and  $\hat{\lambda}_U = 13.59003$ . The estimated 95%

lower and upper confidence limits for true power are given by

$$\widehat{\text{power}}_L = 1 - F(4.085 | 1, 40, 0.000000) = 0.05 \text{ and}$$

$$\widehat{\text{power}}_U = 1 - F(4.085 | 1, 40, 13.59003) = 0.9491.$$

Using the estimate for power, a researcher would conclude that the test was under-powered since this estimated value is below 0.8 (Cohen 1988). Notably, using the point estimate to represent true power is uninformative since the confidence limits have true power anywhere from 0.05 to 0.9491. The confidence interval here is similar to the confidence limits in Figure 4.1.d for the two-sample symmetric case with  $\sigma = 20.0$  at a sample size of 8. Figure 4.5.c likens the clients observations to the a case where discerning treatment groups as different is difficult.

When determining the power of a test *post hoc*, power is not calculated but estimated. For researchers looking to resolve power after observations have been collected, confidence limits for true power can help quantify the uncertainty inherent in a *post hoc* power calculation. Examples 1 and 2, along with the plots in the results section, show a high degree of variability when the null hypothesis cannot be rejected. Hence, researchers are recommended to incorporate power analyses in the *planning* stages of their studies, as opposed to waiting until data have been collected.

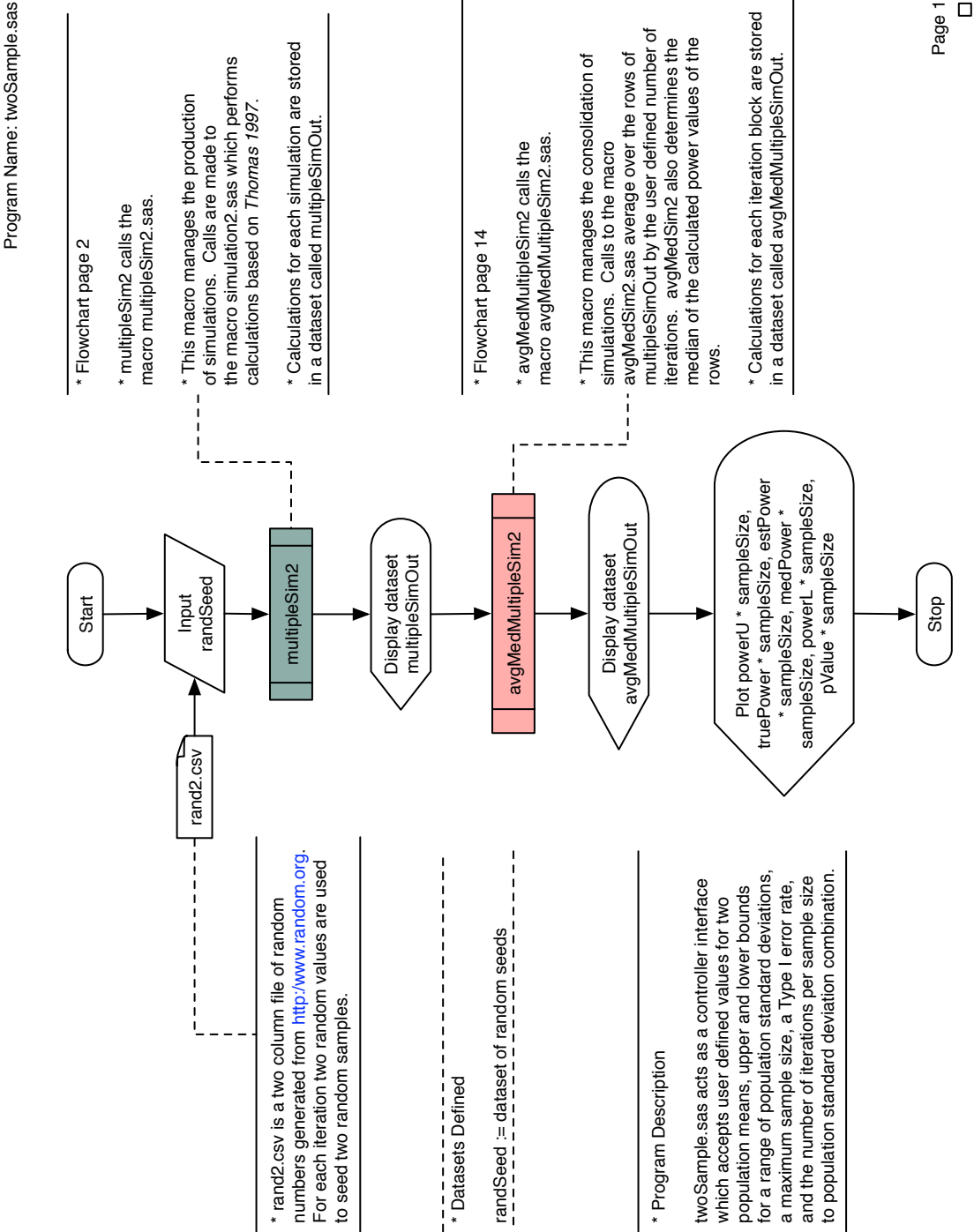


## References

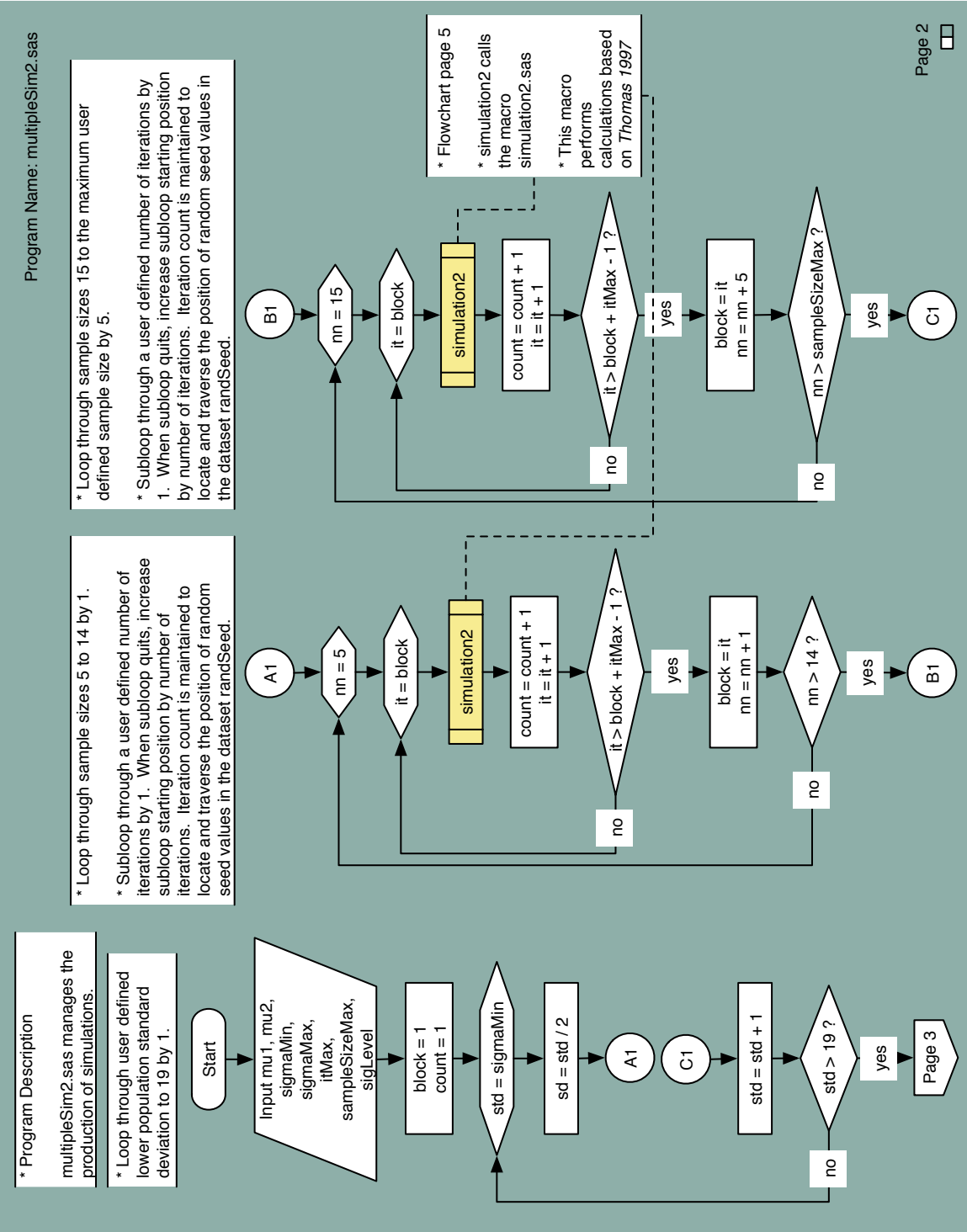
- American Statistical Association (1999), 'Ethical Guidelines for Statistical Practice'.
- Cohen, J. (1988), *Statistical Power Analysis for the Behavioral Sciences*, Lawrence Erlbaum Associates, Inc., 365 Broadway, Hillsdale, New Jersey 07642.
- Gerard, P. D., Smith, D. R. and Weerakkody, G. (1998), 'Limits of Retrospective Power Analysis', *The Journal of Wildlife Management* **62**(2), 801-807.
- Haahr, M. (2009), *Random Integer Generator*, Trinity College, Dublin, Ireland, <http://www.random.org>.
- Hoenig, J. M. and Heisey, D. M. (2001), 'The Abuse of Power: The Pervasive Fallacy of Power Calculations for Data Analysis', *The American Statistician* **55**(1), 19-24.
- Hogarty K. Y. and Kromrey, J. D. (2001), 'Probabilistic Hindsight: A SAS Macro for Retrospective Statistical Power Analysis', *SAS Users Group International Proceedings*, 202-26.
- Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995), *Continuous Univariate Distributions*, Vol. 2 (2nd ed.), John Wiley & Sons, New York, New York.
- Kuehl, R. O. (1999), *Design of Experiments: Statistical Principles of Research Design and Analysis*, (2nd e.), Duxbury Press.
- Lenth, R. V. (2001), 'Some Practical Guidelines for Effective Sample Size Determination', *The American Statistician* **55**(3), 187-193.
- MATLAB (2009), *fminbnd*, The MathWorks, <http://www.mathworks.com/access/helpdesk/help/techdoc/ref/fminbnd.html>.
- Milliken, G. A. and Johnson, D. E (2009), *Analysis of Messy Data, Volume I: Designed Experiments*, (2nd ed.), Chapman & Hall/CRC.
- O'Brien, R. G. (1988), 'Review: [untitled]', *The American Statistician* **42**(4), 266-270.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (2007), *Numerical Recipes: The Art of Scientific Computing*, (3rd ed.), Cambridge University Press.

- R Development Core Team (2009), *optimize*, R Foundation for Statistical Computing, Vienna, Austria, <http://www.R-project.org>.
- SAS Institute Inc. 2006a. SAS® 9.1.3 Language Reference: Dictionary, Fifth Edition, Volumes 1-4 page 574. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2006b. SAS® 9.1.3 Language Reference: Dictionary, Fifth Edition, Volumes 1-4 page 434-447 . Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2004a. SAS® 9.1 User's Guide, page 1929. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2004b. SAS® 9.1 User's Guide, page 3411. Cary, NC: SAS Institute Inc.
- Steidl, R. J., Hayes, J. P. and Schaubert, E. (1997), 'Statistical Power Analysis in Wildlife Research', *The Journal of Wildlife Management* **61**(2), 270-279.
- Taylor, D. J. and Muller, K. E. (1995), 'Computing Confidence Bounds for Power and Sample Size of the General Linear Univariate Model', *The American Statistician* **49**(1), 43-47.
- Taylor, D. J. and Muller, K. E. (1996), 'Bias in linear model power and sample size calculation due to estimating noncentrality', *Communications in Statistics - Theory and Methods* **25**(7), 1595-1610.
- Thomas, L. (1997), 'Retrospective Power Analysis', *Conservation Biology* **11**(1), 276-280.
- Wright, S. P. and O'Brien, R. G. (1988). 'Power analysis in an enhanced GLM procedures: what is might look like. *SAS Users Group International Proceedings* 13, 1097-1102.
- Yuan, K.-H. and Maxwell, S. (2005), 'On the Post Hoc Power in Testing Mean Differences', *Journal of Educational and Behavioral Statistics* **30**(2), 141-167.
- Zumbo, B. D. and Hubley, A. M. (1998), 'A Note on Misconceptions Concerning Prospective and Retrospective Power', *Journal of the Royal Statistical Society. Series D (The Statistician)* **47**(2), 385--388.

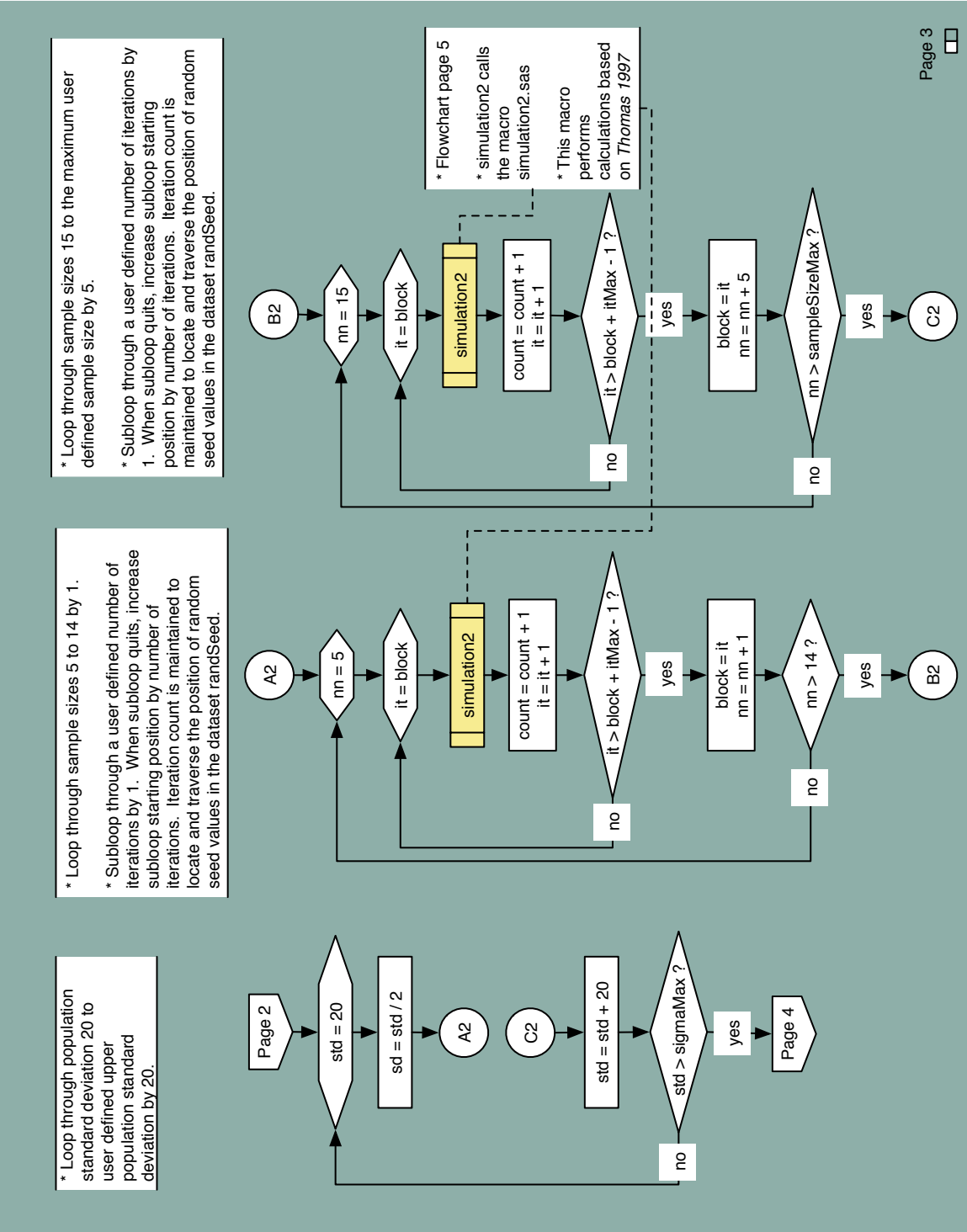
Appendix A: Two-Sample Flow Chart of Power Simulations



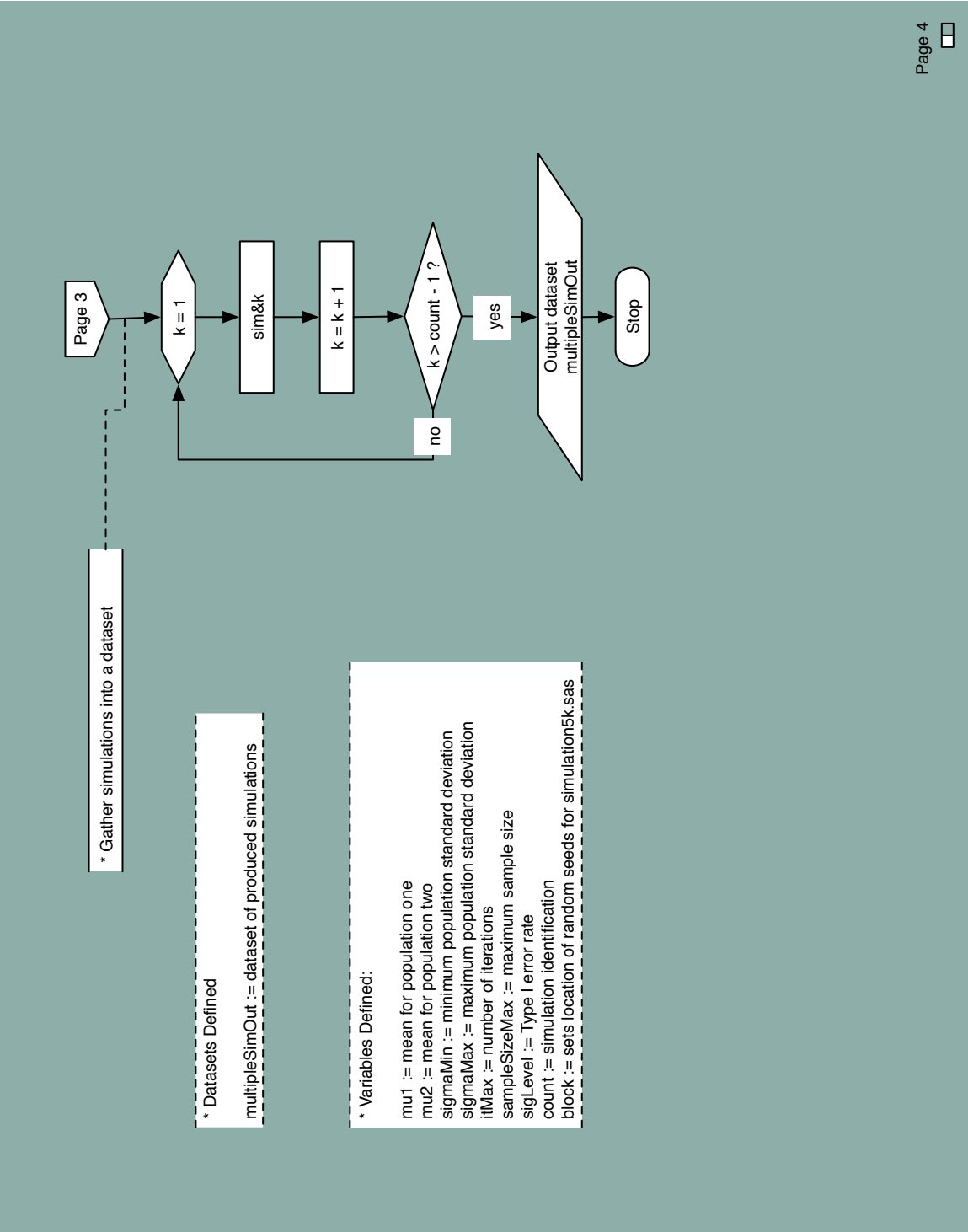
Appendix A: Two-Sample Flow Chart of Power Simulations



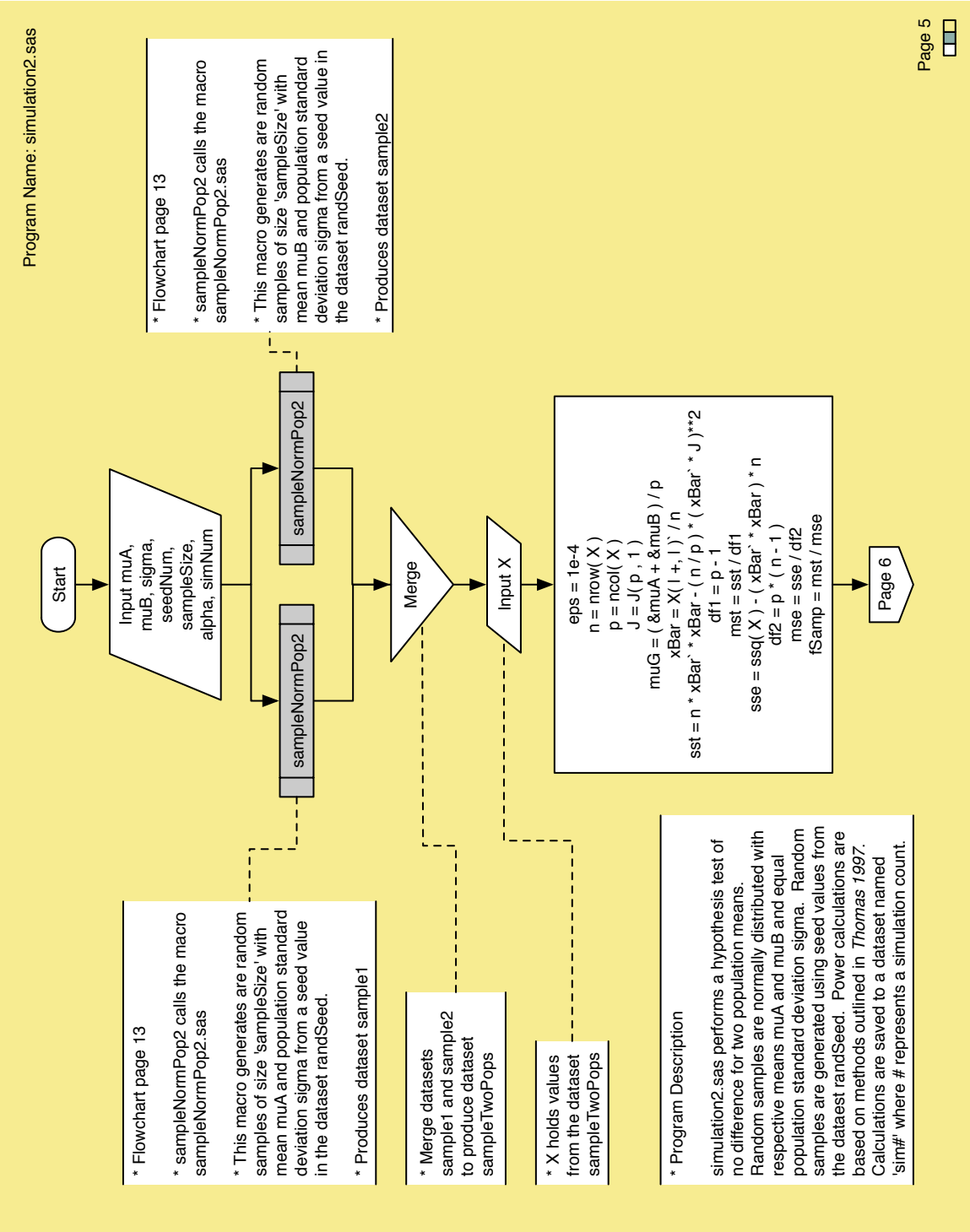
Appendix A: Two-Sample Flow Chart of Power Simulations



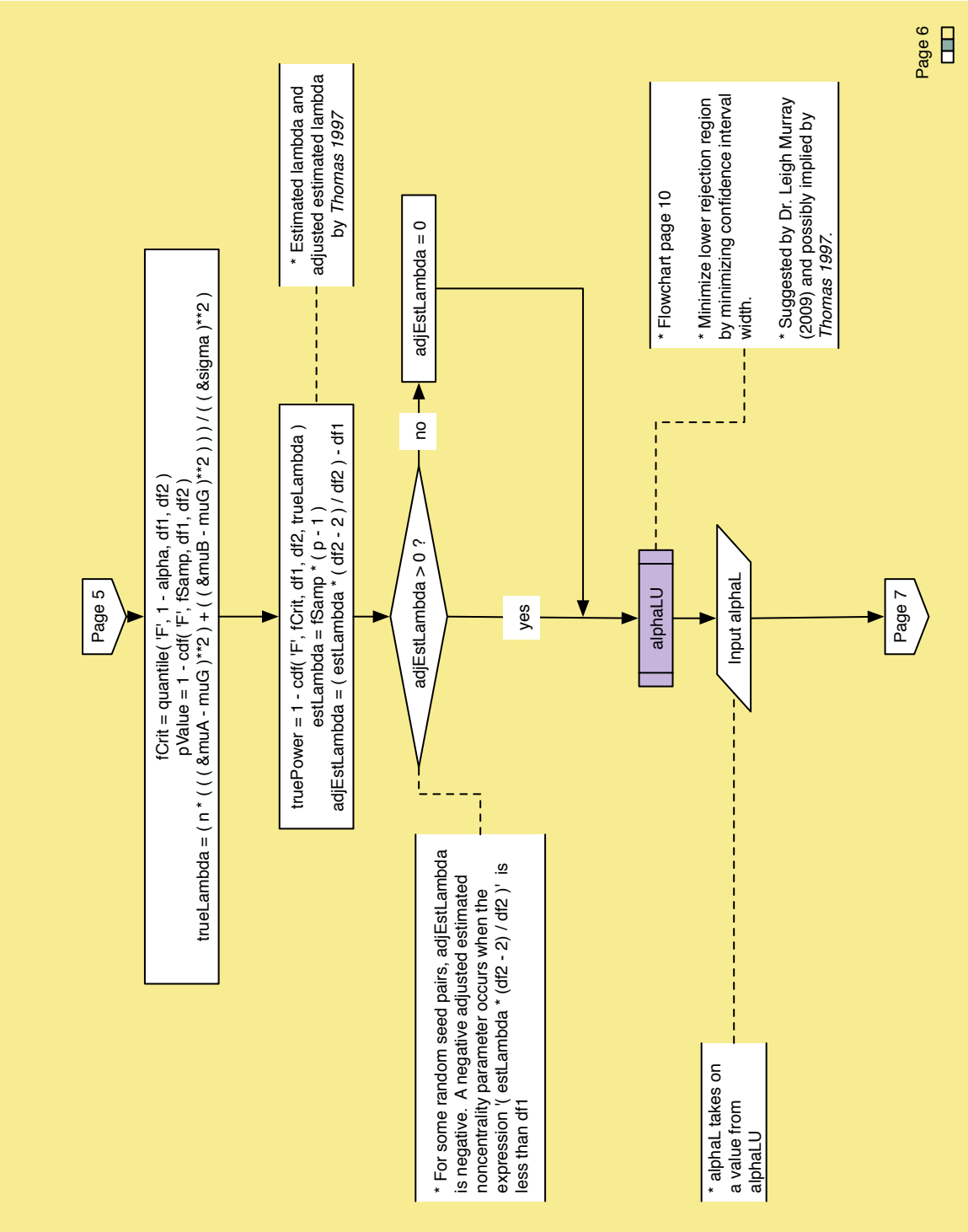
Appendix A: Two-Sample Flow Chart of Power Simulations



Appendix A: Two-Sample Flow Chart of Power Simulations

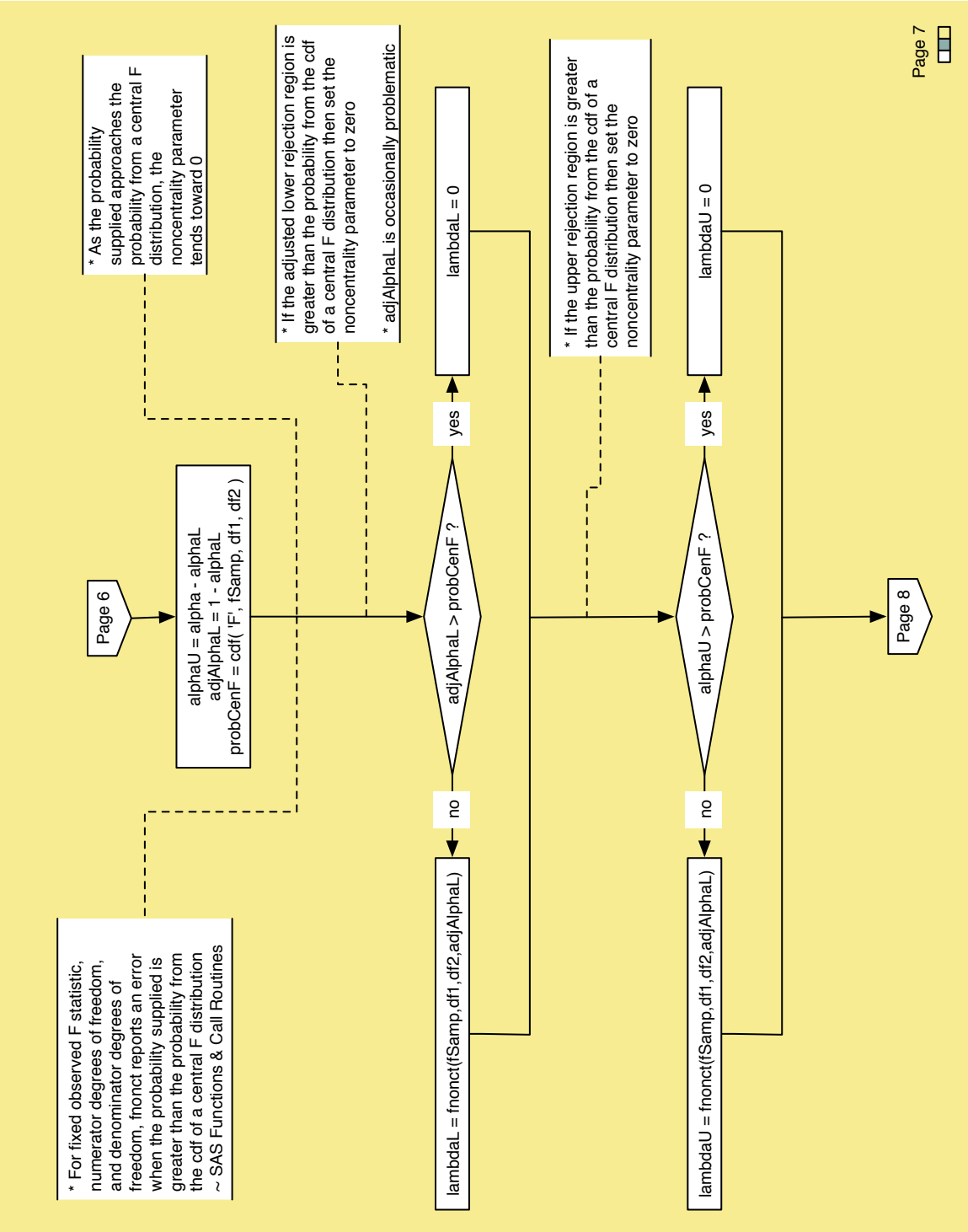


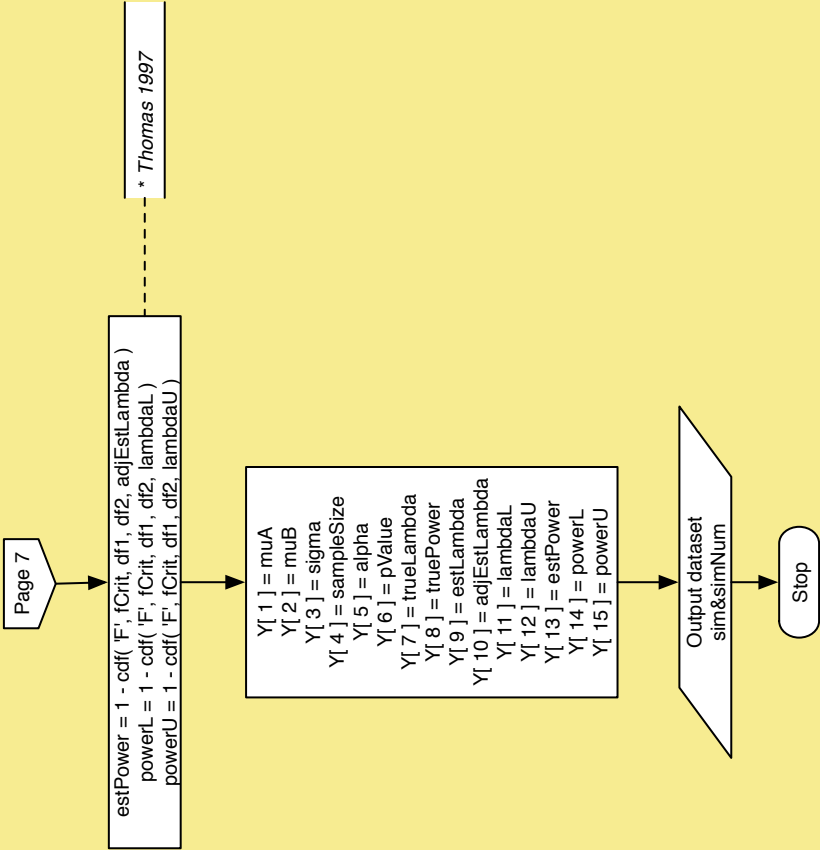
Appendix A: Two-Sample Flow Chart of Power Simulations





Appendix A: Two-Sample Flow Chart of Power Simulations

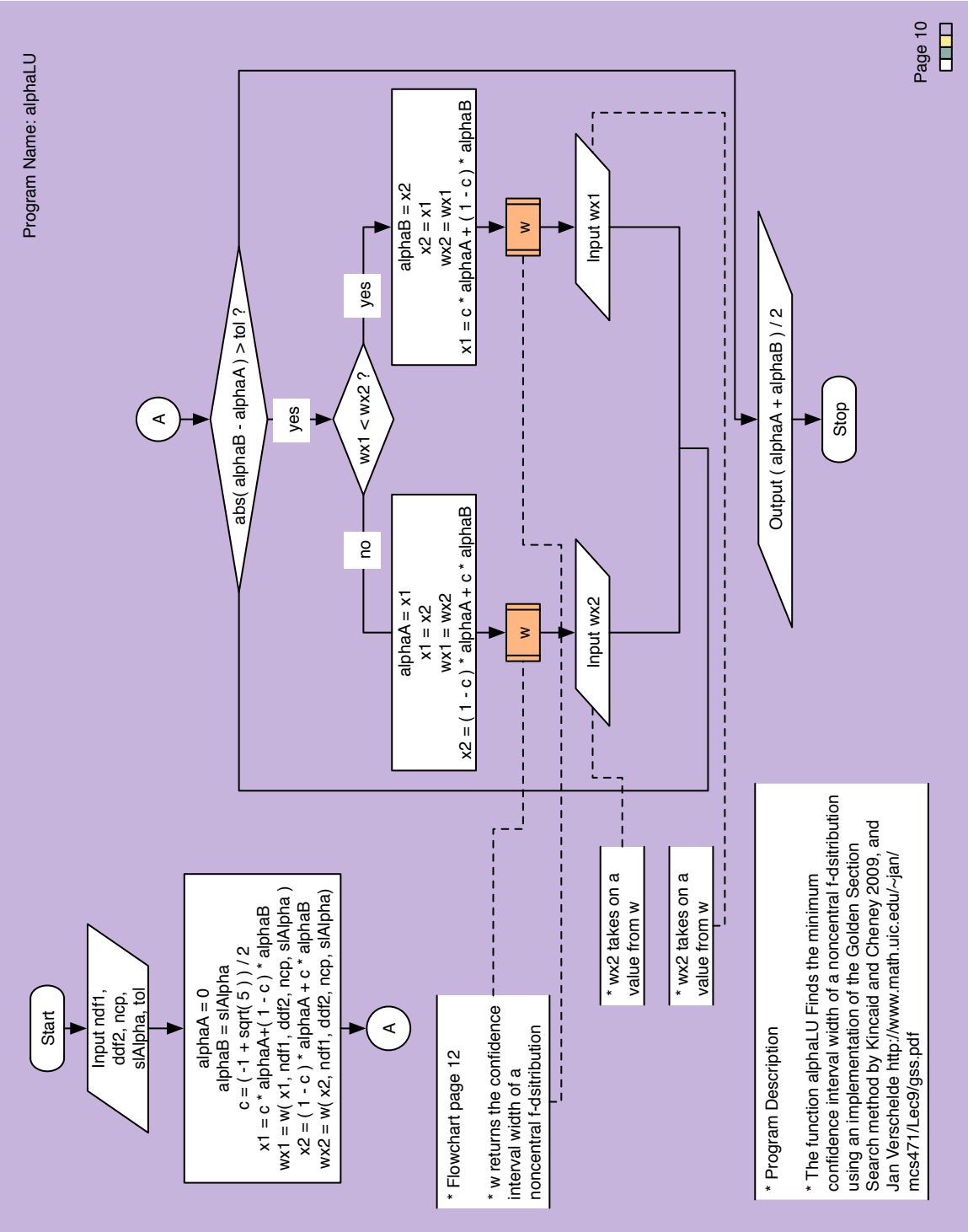




Appendix A: Two-Sample Flow Chart of Power Simulations

* Variables Defined	* Datasets Defined
muA := mean for population one	sampleTwoPops := random samples dataset
muB := mean for population two	sim&simNum := simulation dataset
sigma := population standard deviation	
seedNum := location of random seeds	
sampleSize := sample size	
alpha := level of significance	
simNum := simulation identification	
X := samples from normal populations	
eps := tolerance for Golden Section Search methods	
n := number rows	
p := number columns	
xBar := sample mean	
sst := sum of squares for treatment	
df1 := numerator degrees of freedom	
mst := mean square for treatment	
sse := sum of squares for error	
df2 := denominator degrees of freedom	
mse := mean square for error	
fSamp := observed F statistic	
fCrit := critical value from a central F distribution	
pValue := p-value for the test	
trueLambda := noncentrality parameter	
truePower := power for the test	
estLambda := calculated noncentrality parameter	
adjEstLambda := adjusted calculated noncentrality parameter	
alphaL := optimized lower critical region	
alphaU := optimized upper critical region	
probCenF := probability of observed F statistic of the central F distribution	
estPower := calculated power	
powerL := lower confidence interval for calculated power	
powerU := upper confidence interval for calculated power	

Appendix A: Two-Sample Flow Chart of Power Simulations



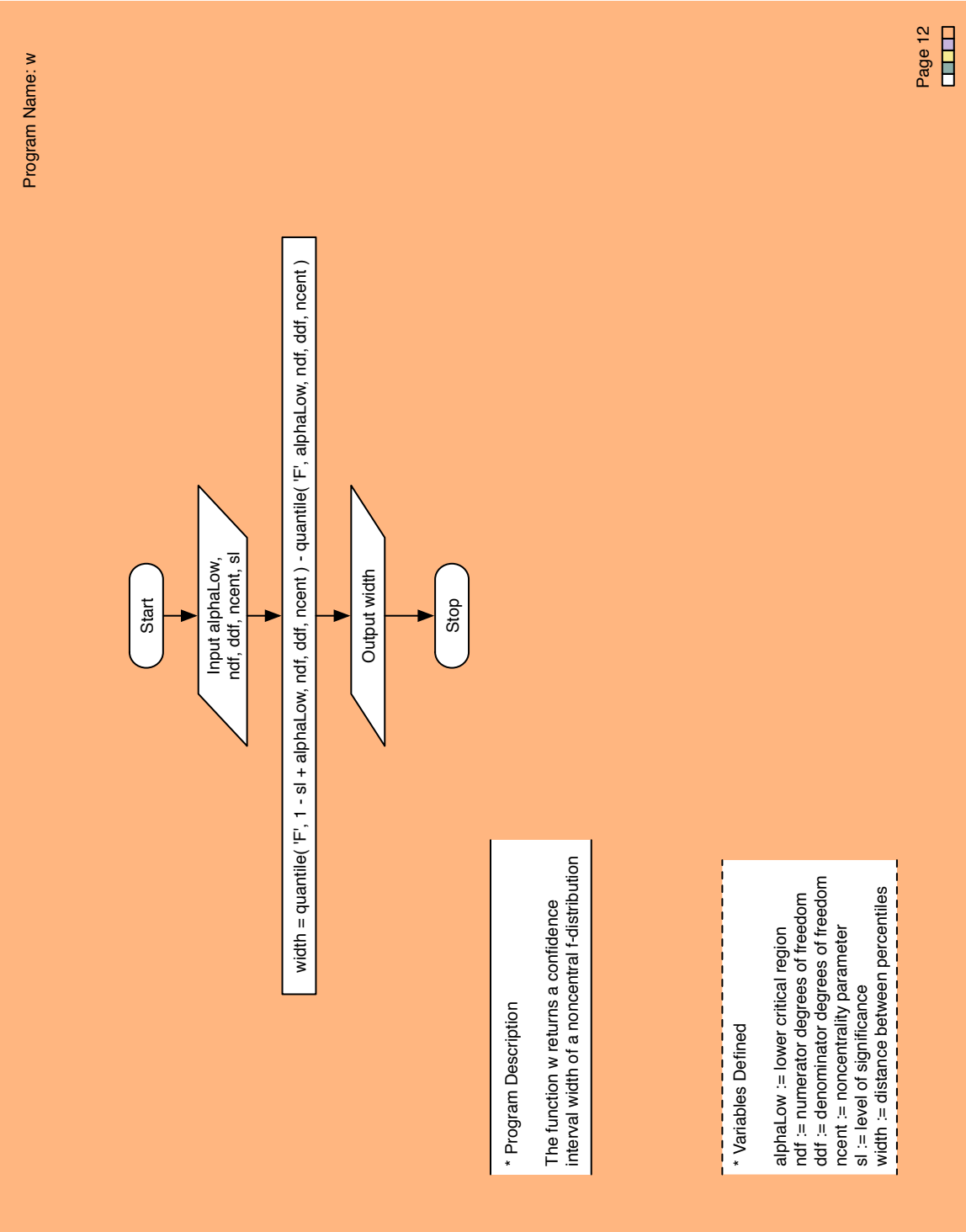
Appendix A: Two-Sample Flow Chart of Power Simulations

\* Variables Defined

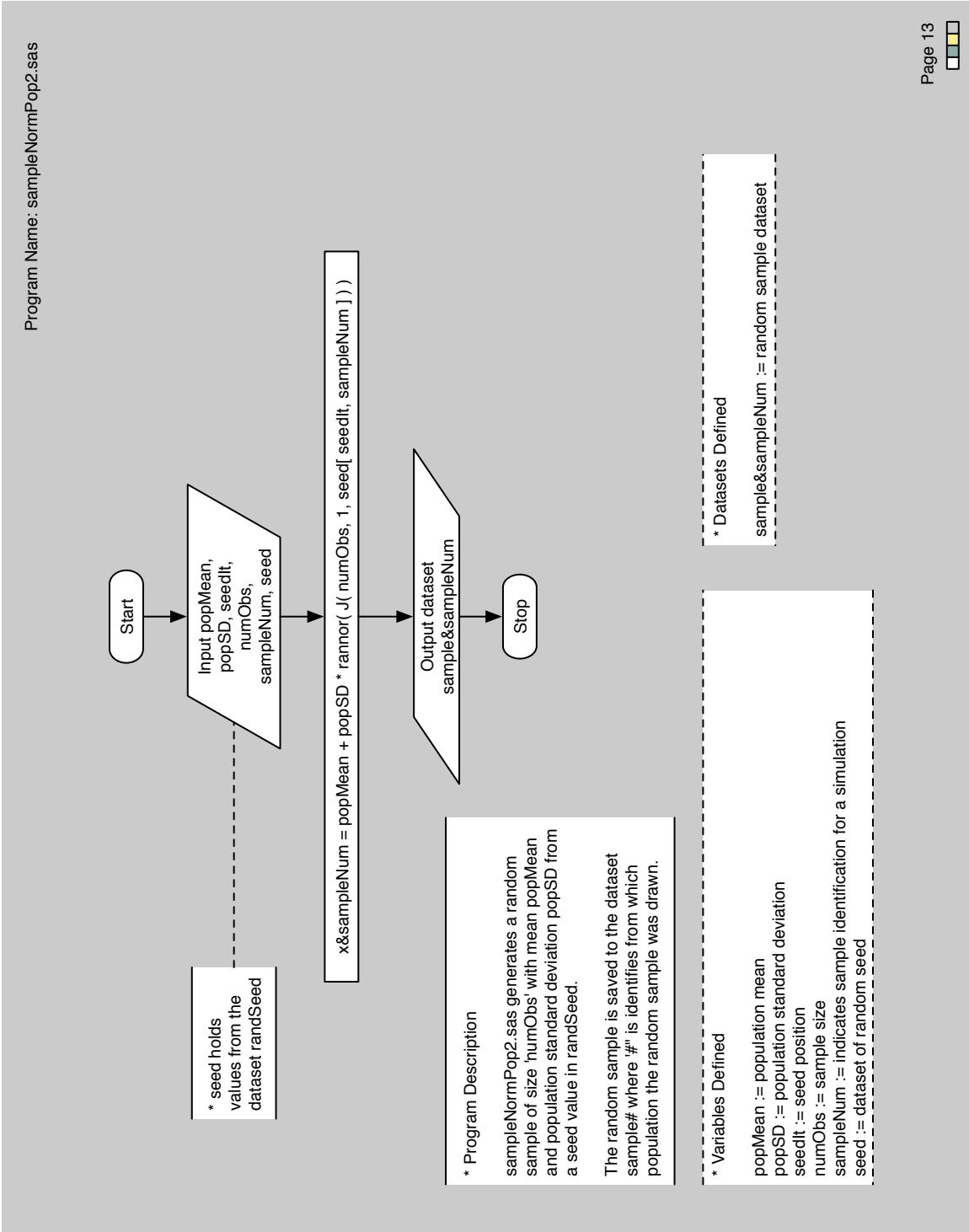
ndf1 := numerator degrees of freedom  
ddf2 := denominator degrees of freedom  
ncp := noncentrality parameter  
slAlpha := level of significance  
tol := tolerance for Golden Section Search method  
alphaA := lower bound for optimized lower alpha  
alphaB := upper bound for optimized lower alpha  
c := Golden ratio constant reduction factor  
x1 := percentile associated with alphaA  
wx1 := height associated with x1  
x2 := percentile associated with alphaB  
wx2 := height associated with x2



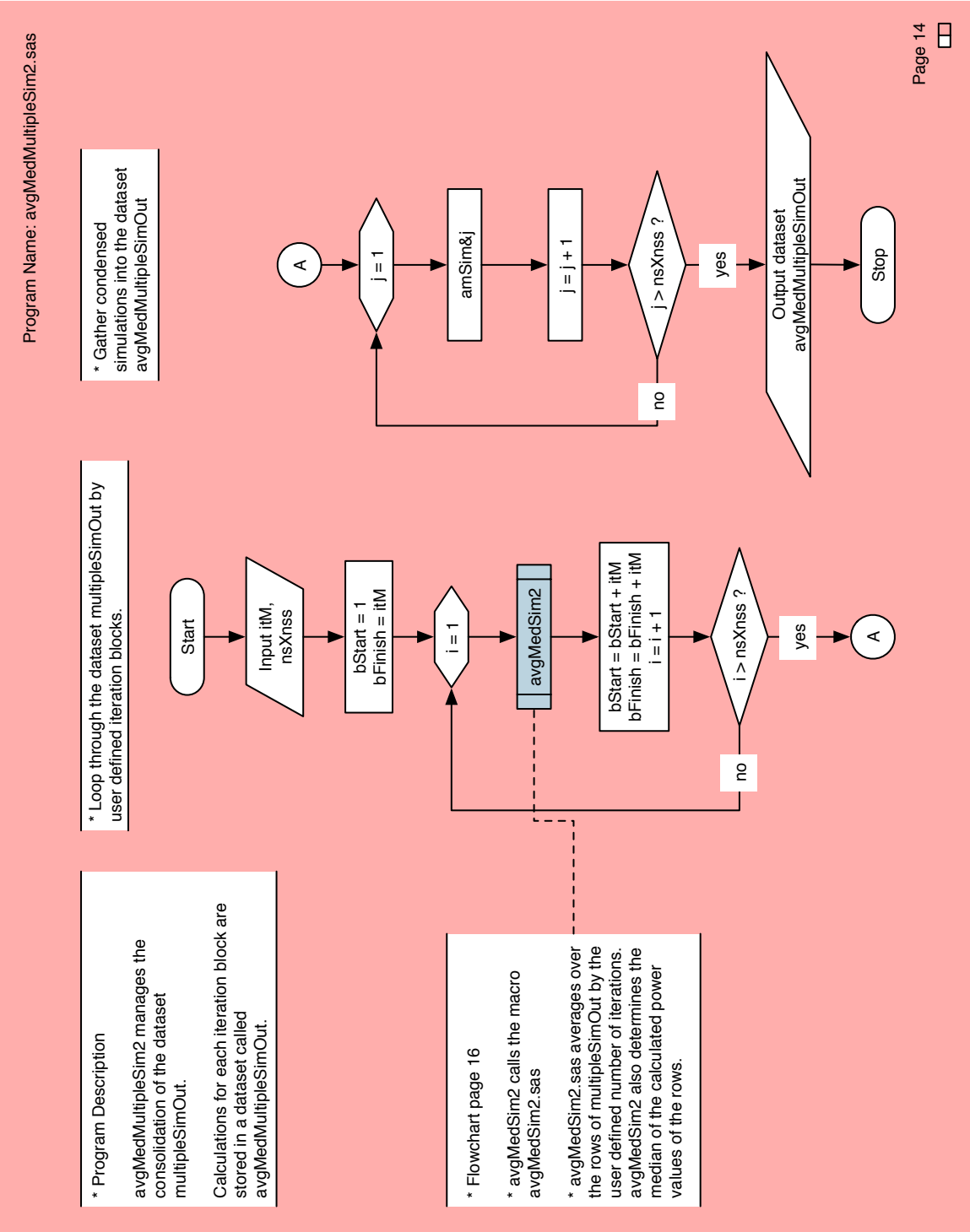
Appendix A: Two-Sample Flow Chart of Power Simulations



Appendix A: Two-Sample Flow Chart of Power Simulations

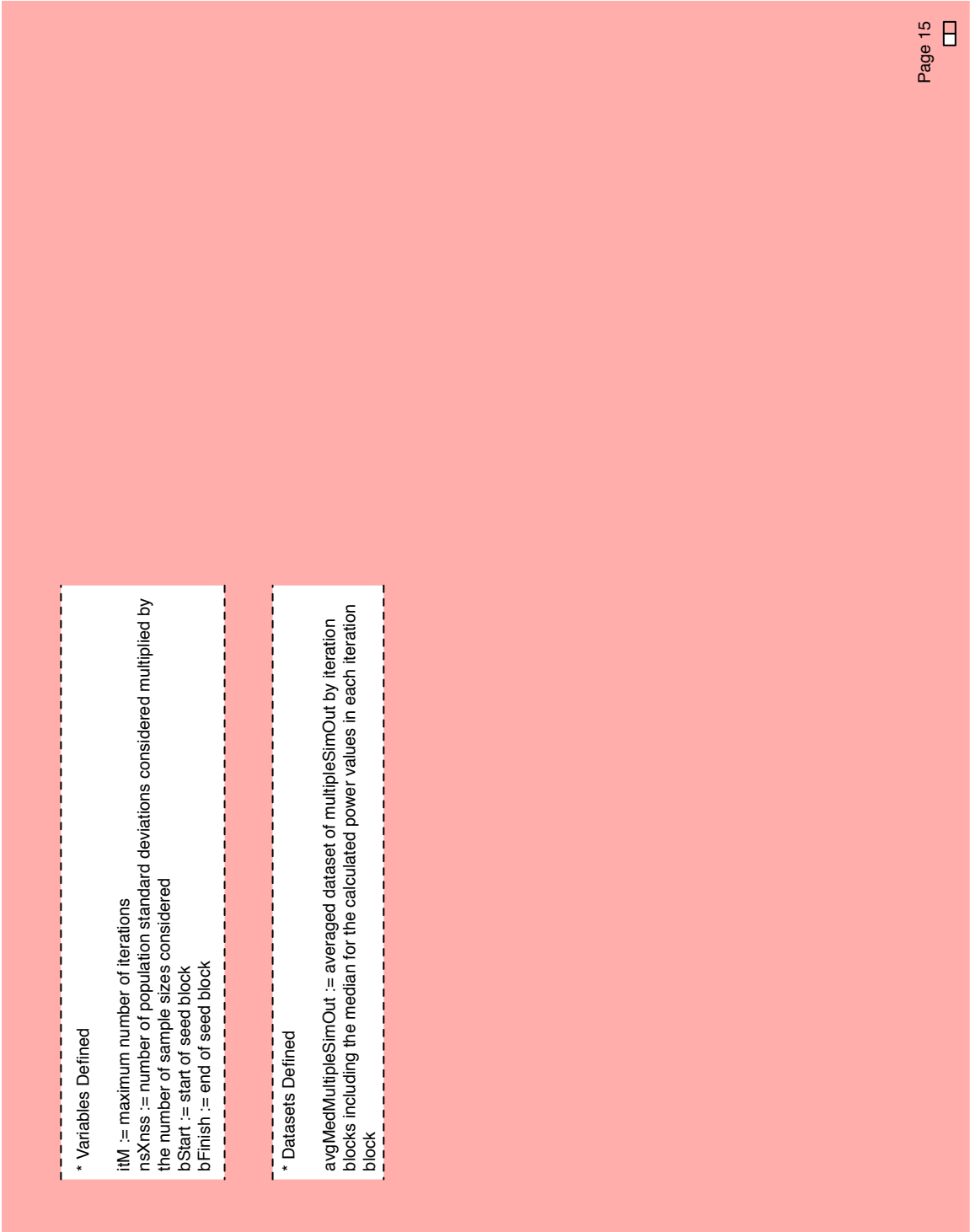


Appendix A: Two-Sample Flow Chart of Power Simulations





Appendix A: Two-Sample Flow Chart of Power Simulations



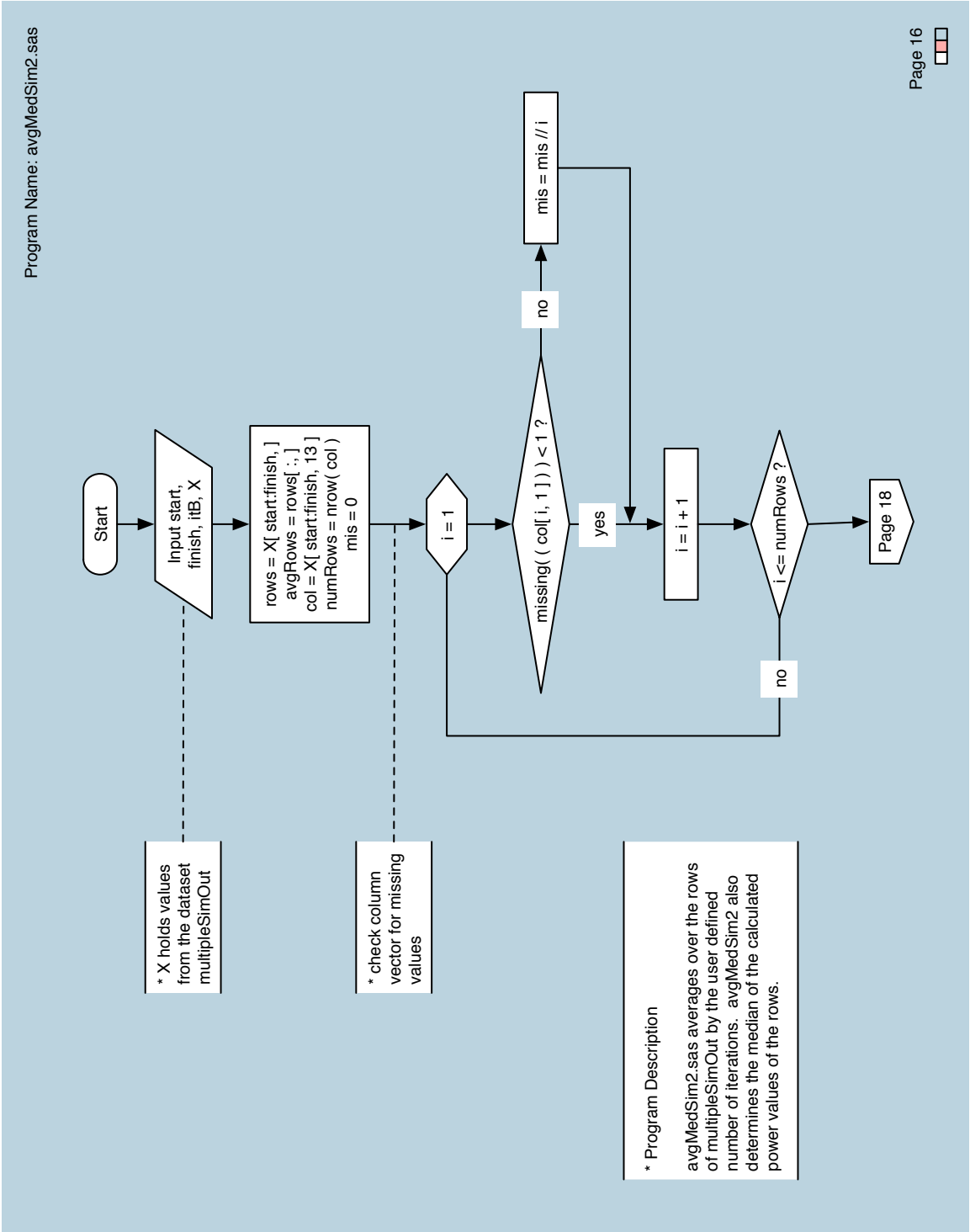
\* Variables Defined

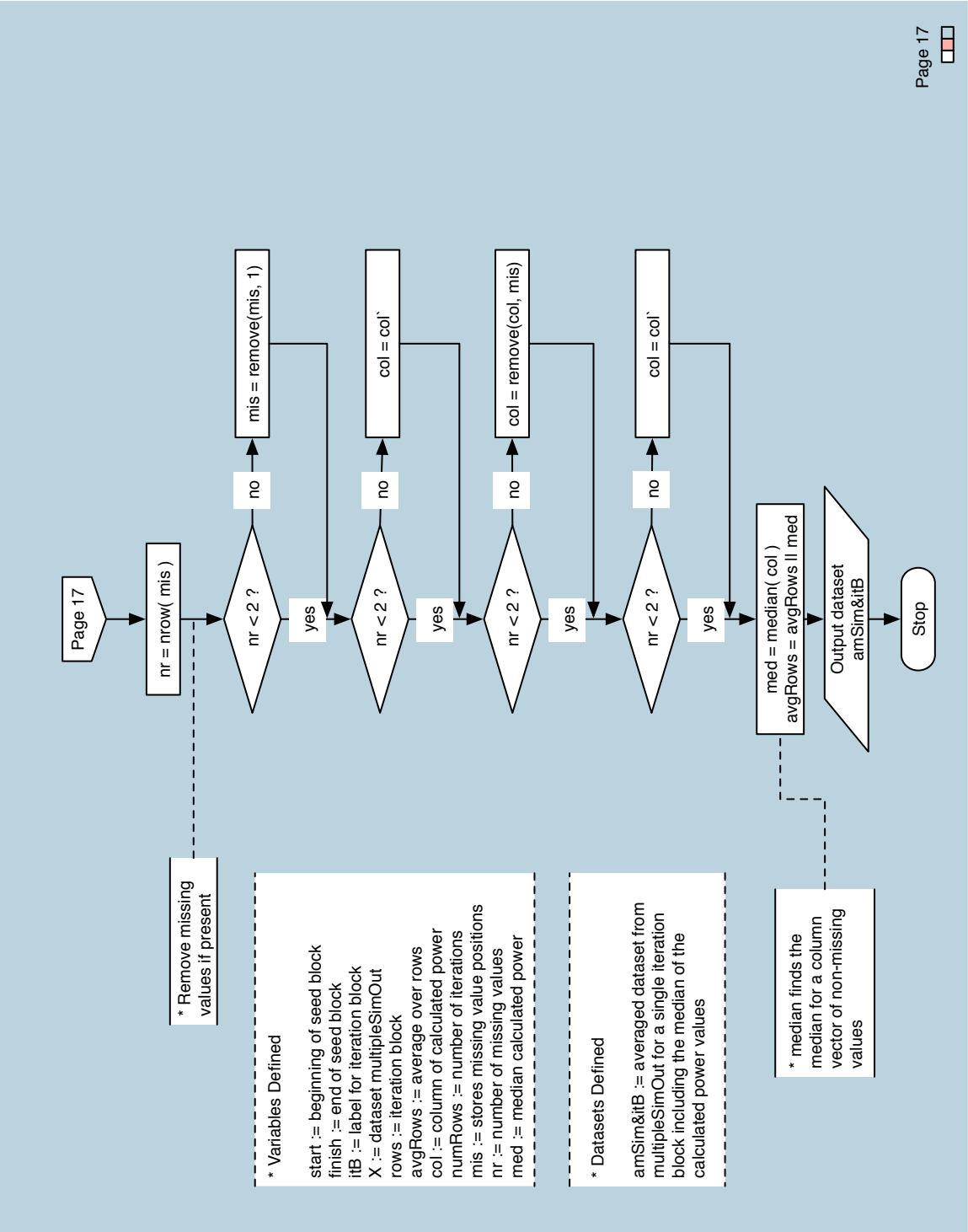
itM := maximum number of iterations  
nsXnss := number of population standard deviations considered multiplied by  
the number of sample sizes considered  
bStart := start of seed block  
bFinish := end of seed block

\* Datasets Defined

avgMedMultipleSimOut := averaged dataset of multipleSimOut by iteration  
blocks including the median for the calculated power values in each iteration  
block

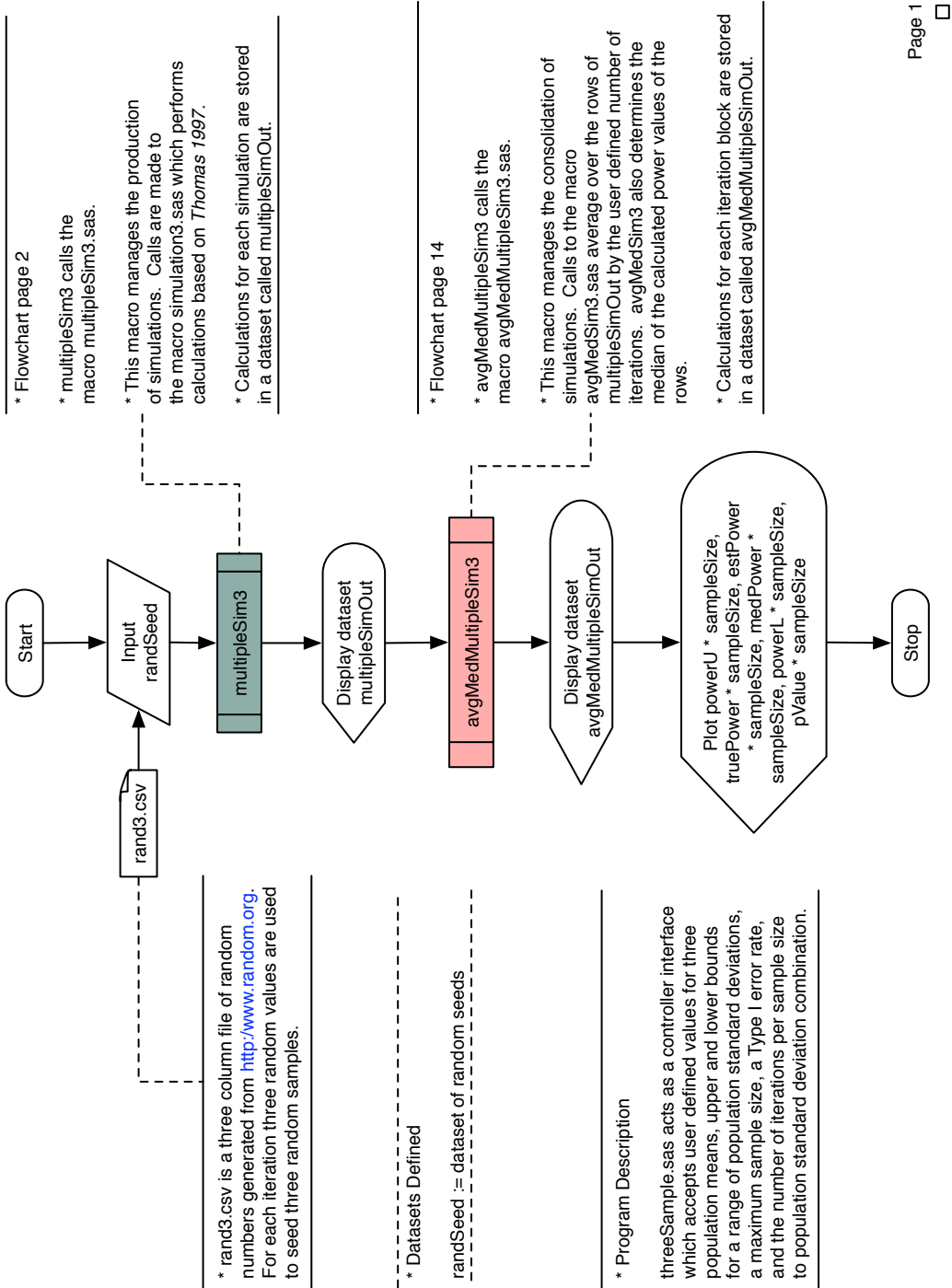
Appendix A: Two-Sample Flow Chart of Power Simulations



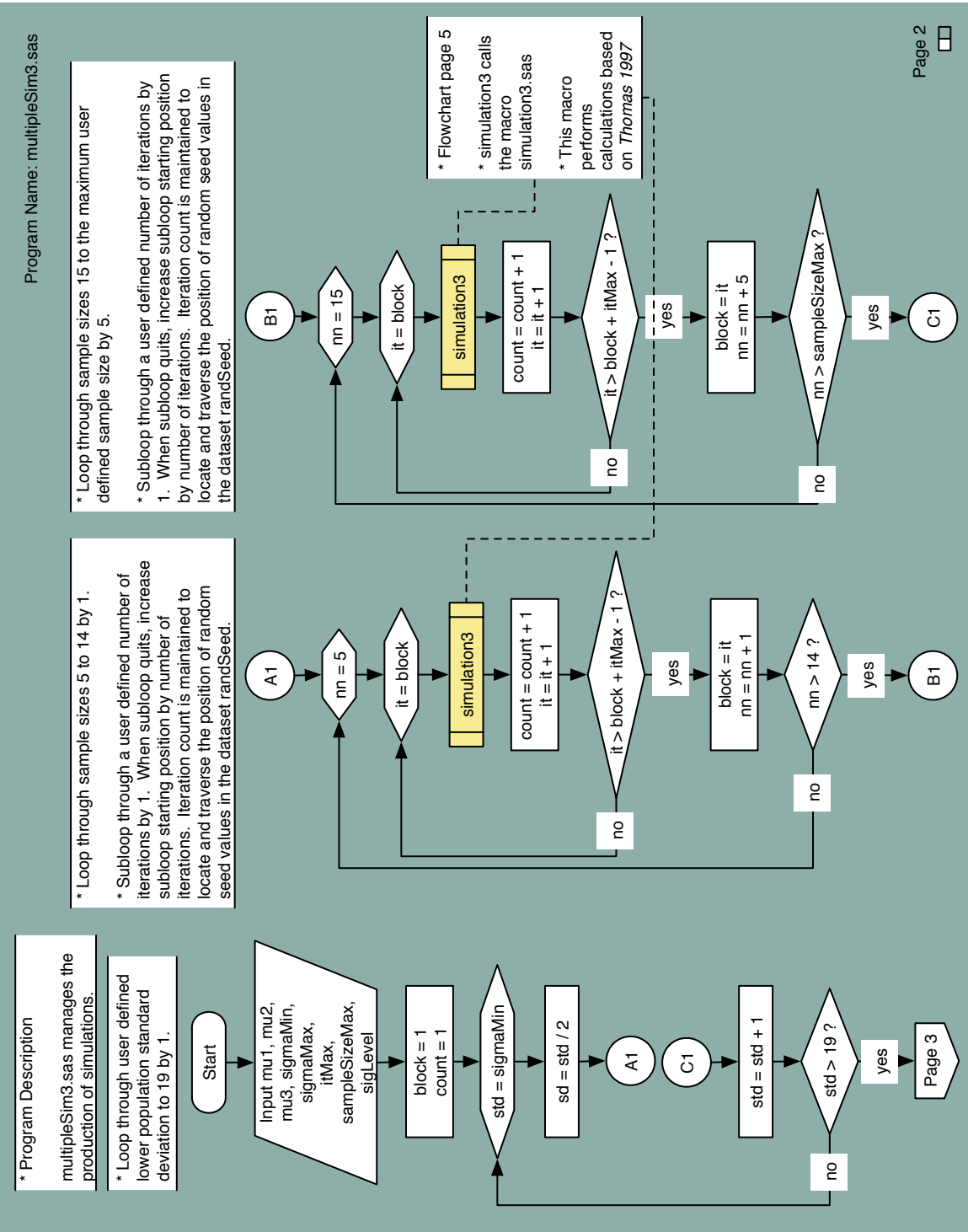


Appendix A: Three-Sample Flow Chart of Power Simulations

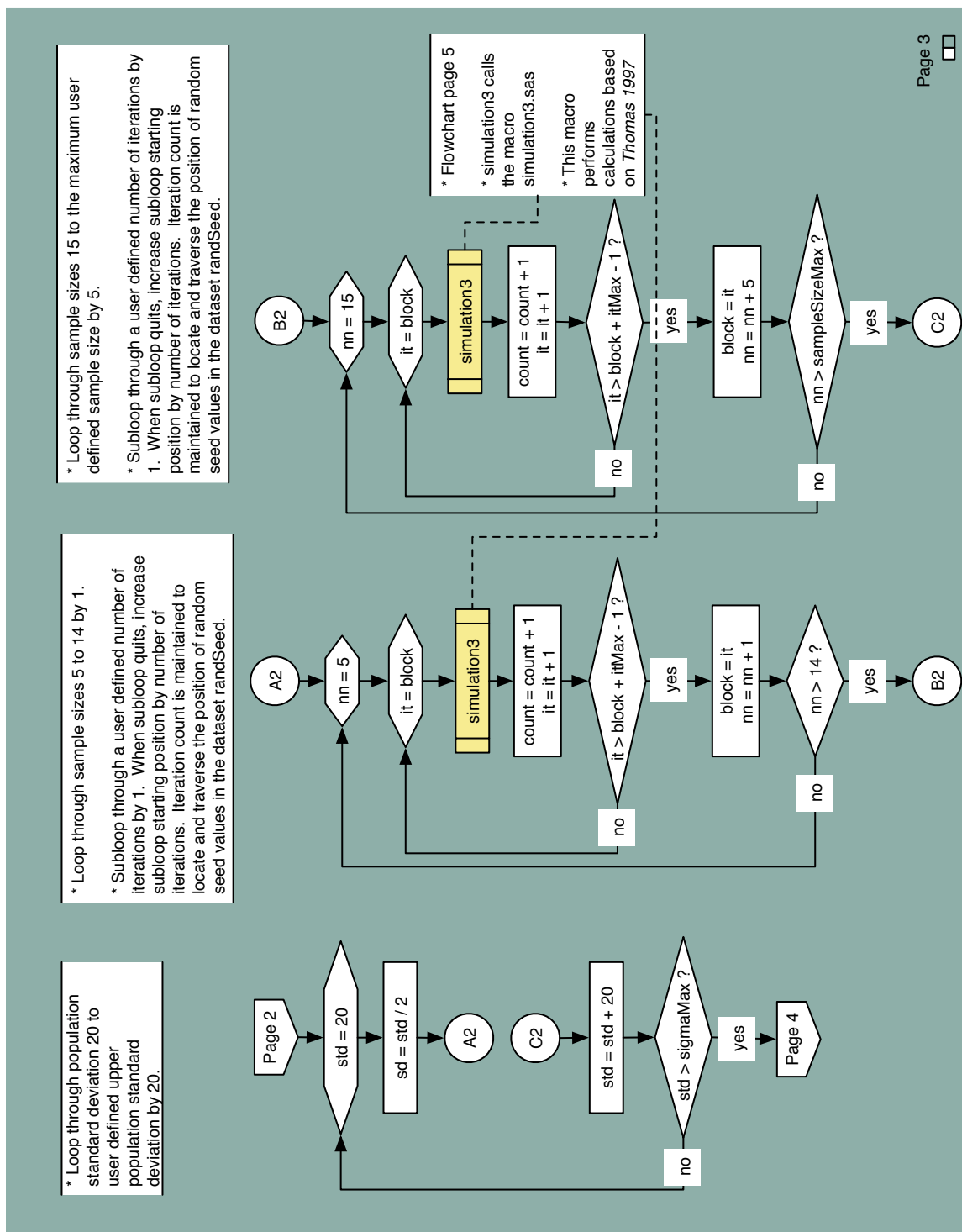
Program Name: threeSample.sas



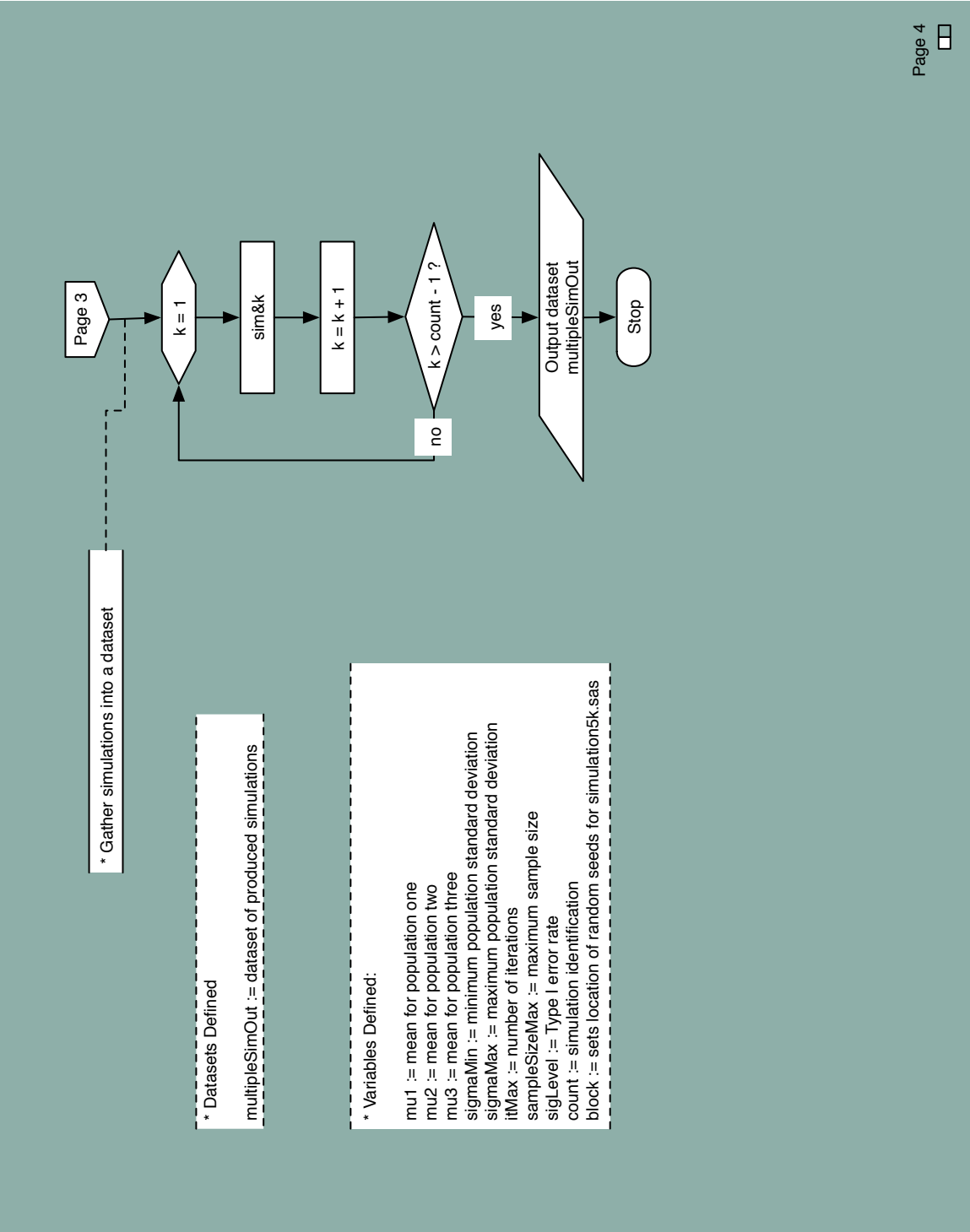
Appendix A: Three-Sample Flow Chart of Power Simulations



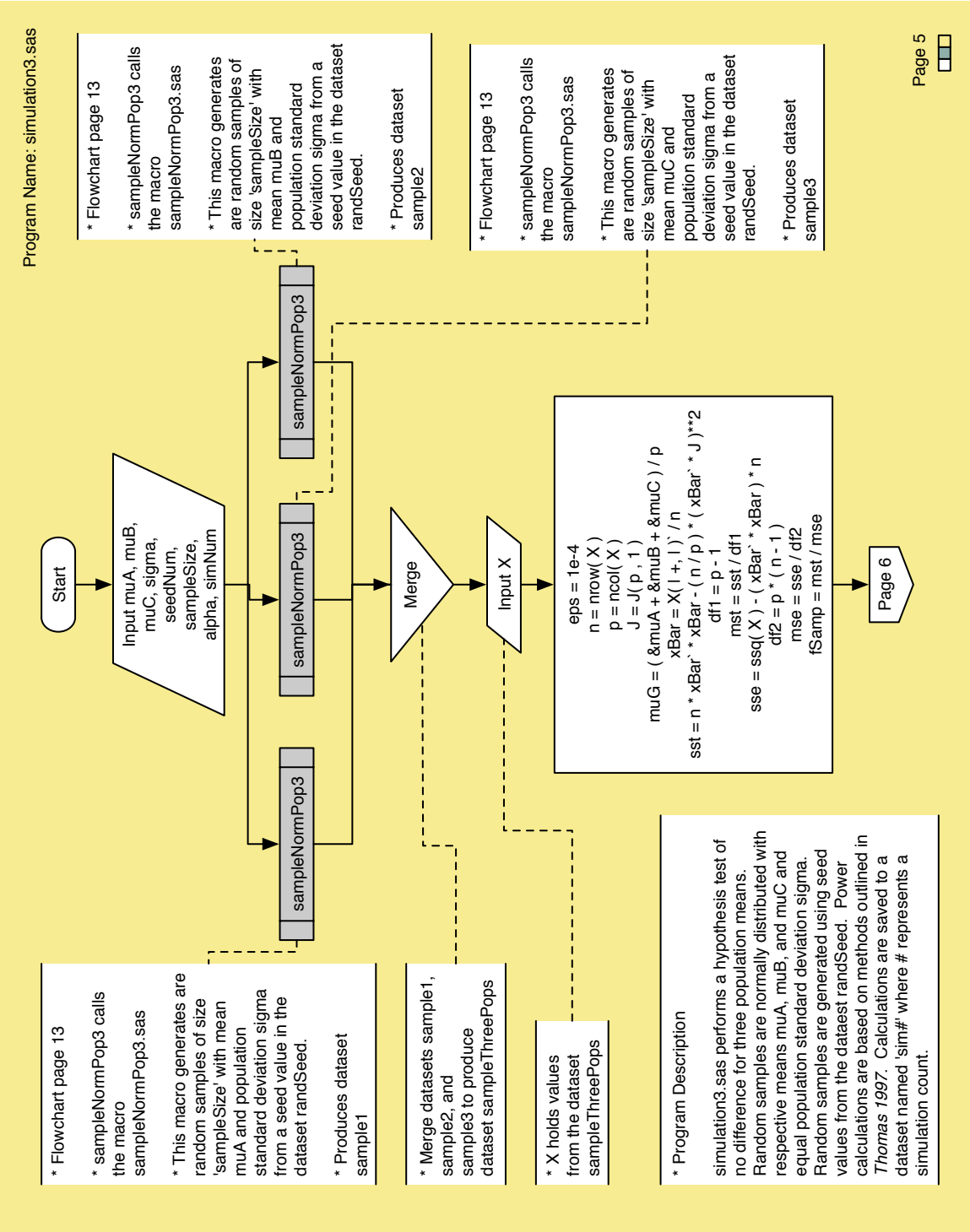
## Appendix A: Three-Sample Flow Chart of Power Simulations



Appendix A: Three-Sample Flow Chart of Power Simulations

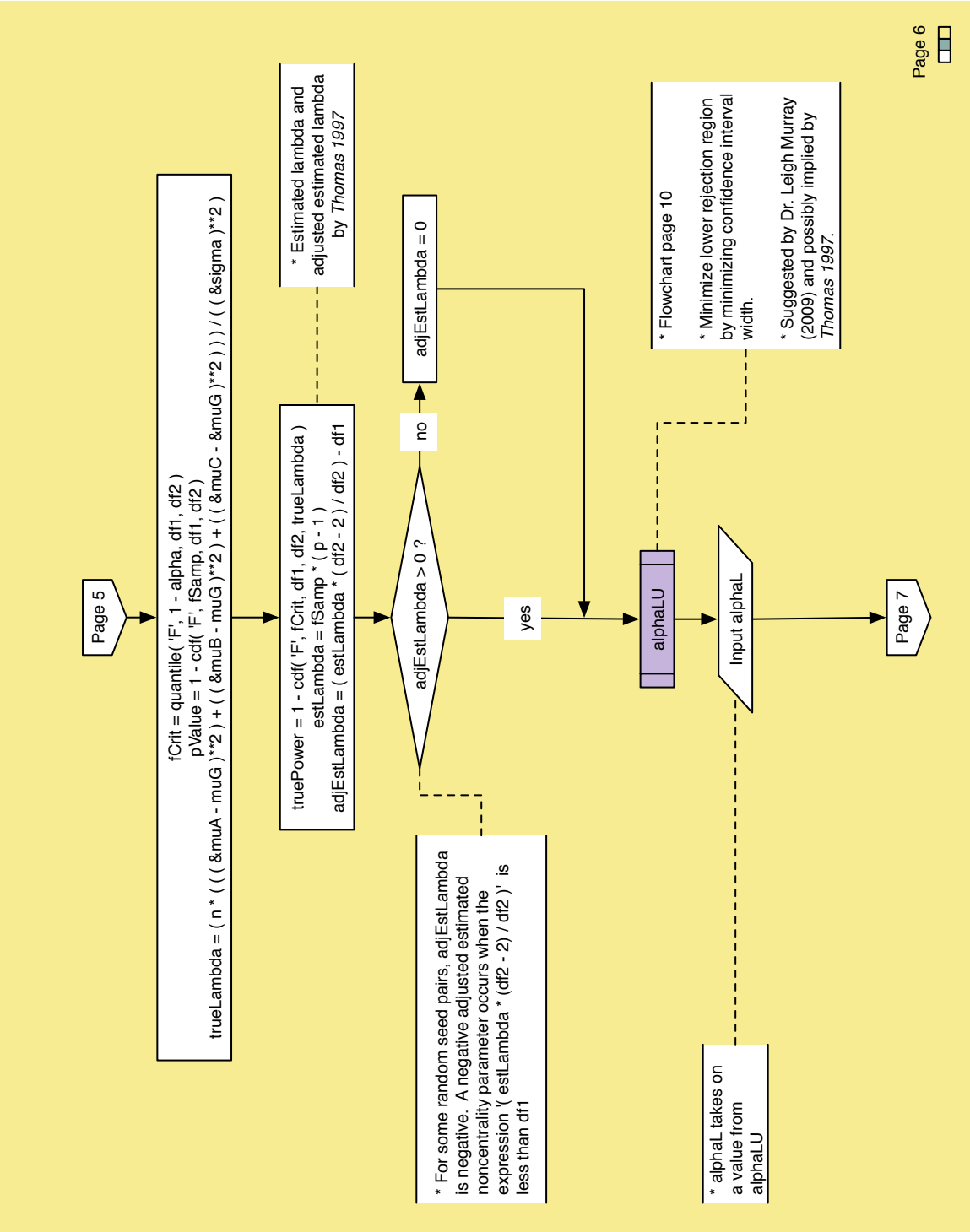


Appendix A: Three-Sample Flow Chart of Power Simulations

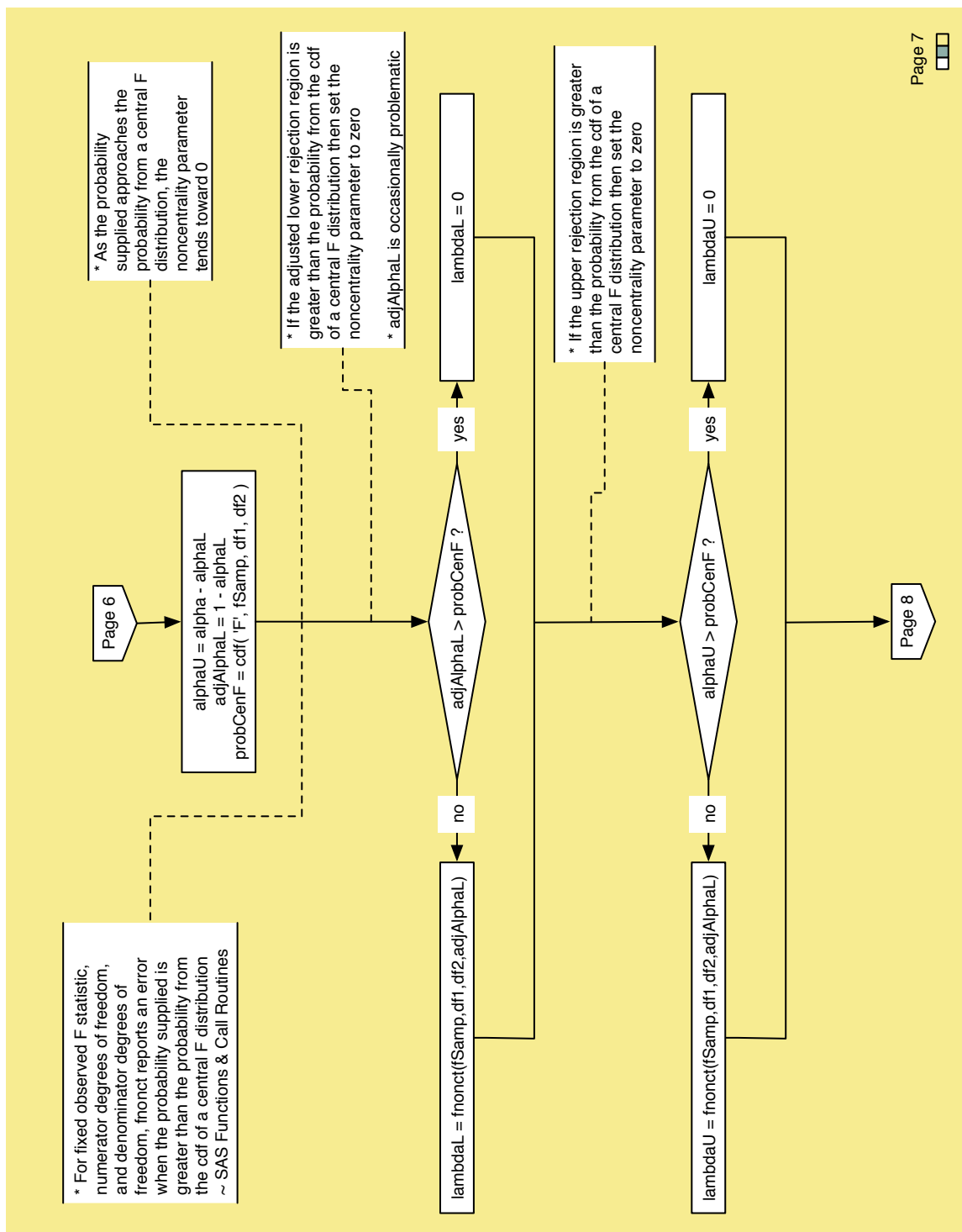




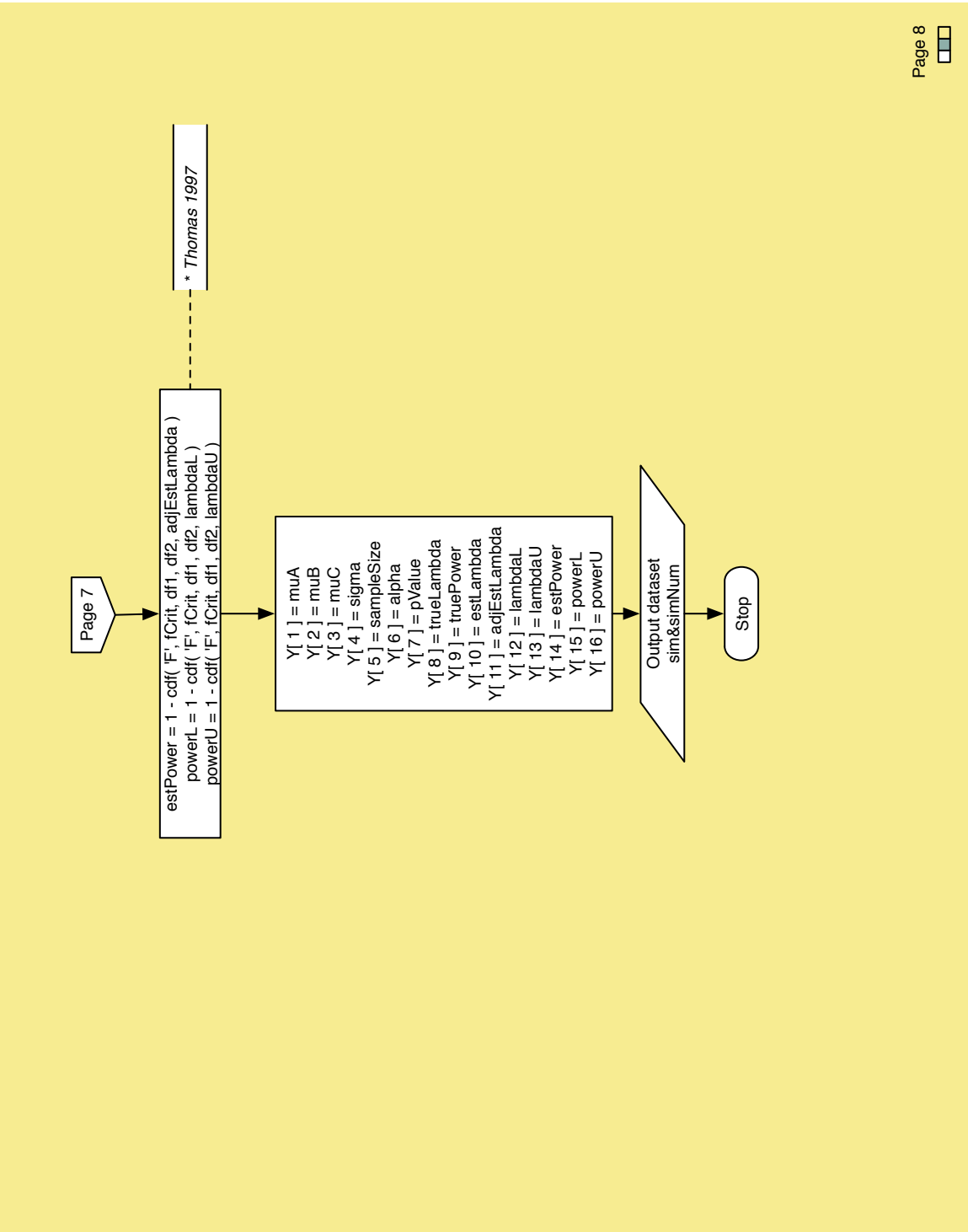
Appendix A: Three-Sample Flow Chart of Power Simulations



## Appendix A: Three-Sample Flow Chart of Power Simulations



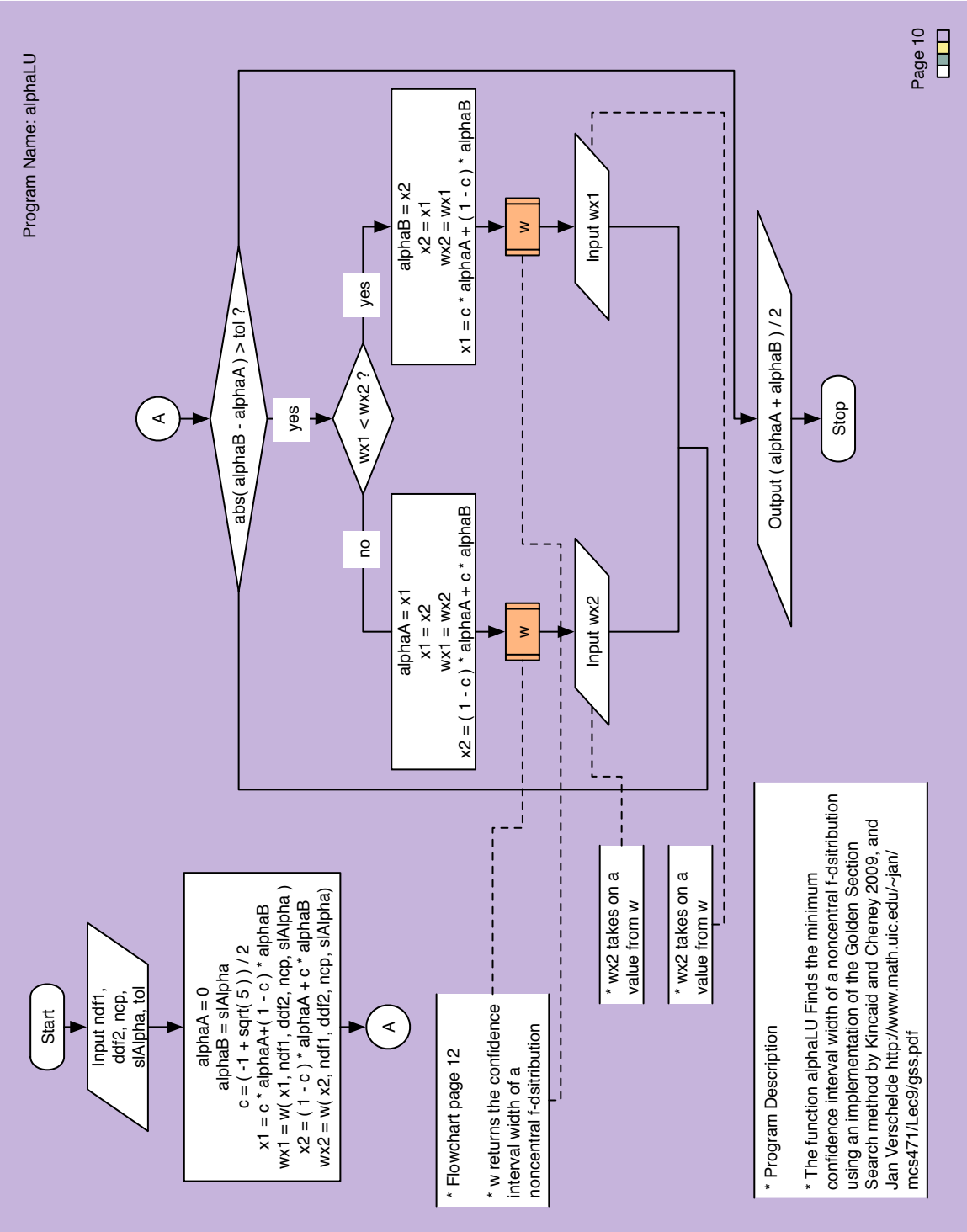
Appendix A: Three-Sample Flow Chart of Power Simulations



Appendix A: Three-Sample Flow Chart of Power Simulations

<div>* Variables Defined</div> <div>muA := mean for population one</div> <div>muB := mean for population two</div> <div>muC := mean for population three</div> <div>muG := average of the tree population means</div> <div>sigma := population standard deviation</div> <div>seedNum := location of random seeds</div> <div>sampleSize := sample size</div> <div>alpha := level of significance</div> <div>simNum := simulation identification</div> <div>X := samples from normal populations</div> <div>eps := tolerance for Golden Section Search methods</div> <div>n := number rows</div> <div>p := number columns</div> <div>xBar := sample mean</div> <div>sst := sum of squares for treatment</div> <div>df1 := numerator degrees of freedom</div> <div>mst := mean square for treatment</div> <div>sse := sum of squares for error</div> <div>df2 := denominator degrees of freedom</div> <div>mse := mean square for error</div> <div>fSamp := observed F statistic</div> <div>fCrit := critical value from a central F distribution</div> <div>pValue := p-value for the test</div> <div>trueLambda := noncentrality parameter</div> <div>truePower := power for the test</div> <div>estLambda := calculated noncentrality parameter</div> <div>adjEstLambda := adjusted calculated noncentrality parameter</div> <div>alphaL := optimized lower critical region</div> <div>alphaU := optimized upper critical region</div> <div>probCenF := probability of observed F statistic of the central F distribution</div> <div>estPower := calculated power</div> <div>powerL := lower confidence interval for calculated power</div> <div>powerU := upper confidence interval for calculated power</div>	<div>* Datasets Defined</div> <div>sampleThreePops := random samples dataset</div> <div>sim&amp;simNum := simulation dataset</div>
---	--

Appendix A: Three-Sample Flow Chart of Power Simulations



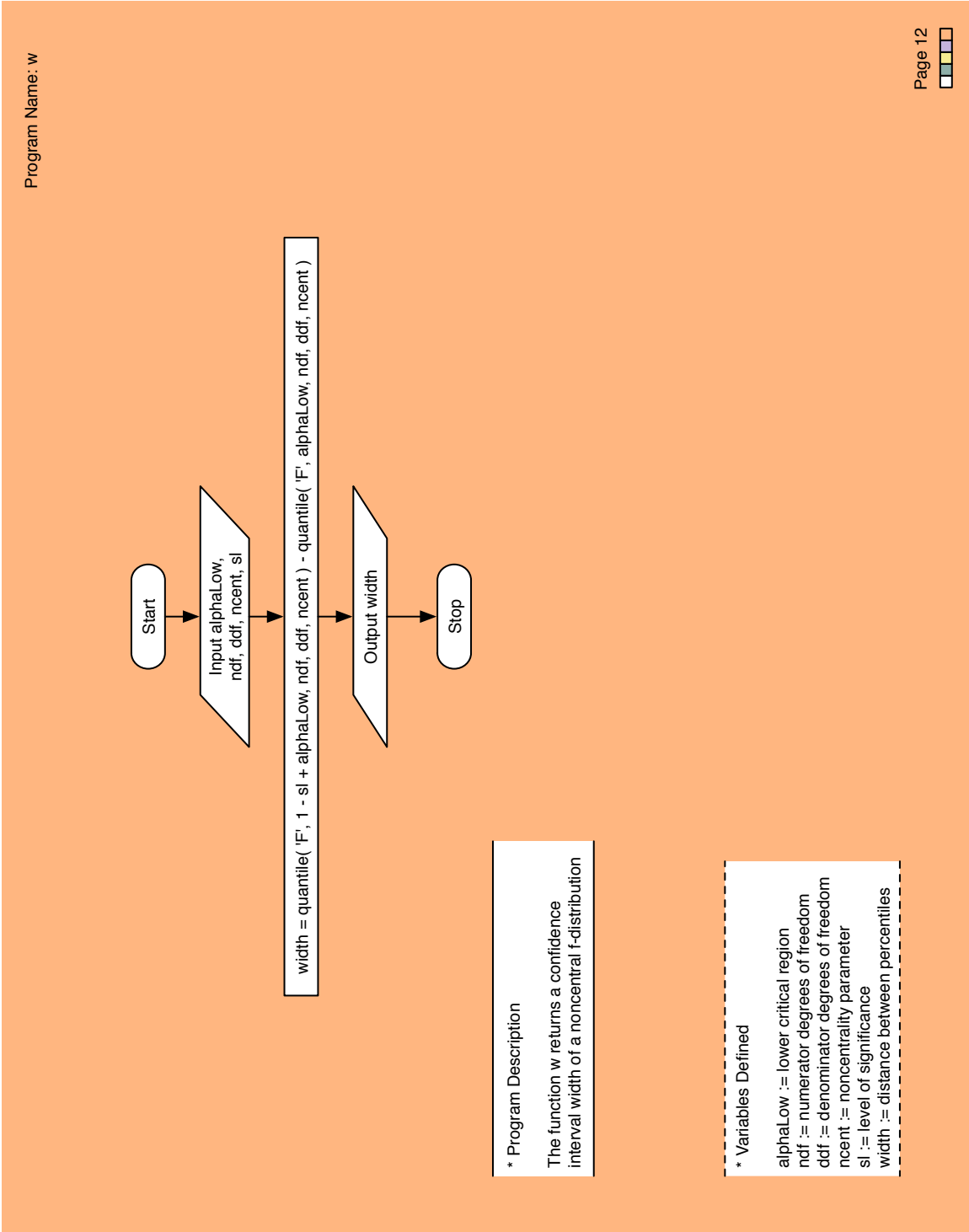
Appendix A: Three-Sample Flow Chart of Power Simulations

\* Variables Defined

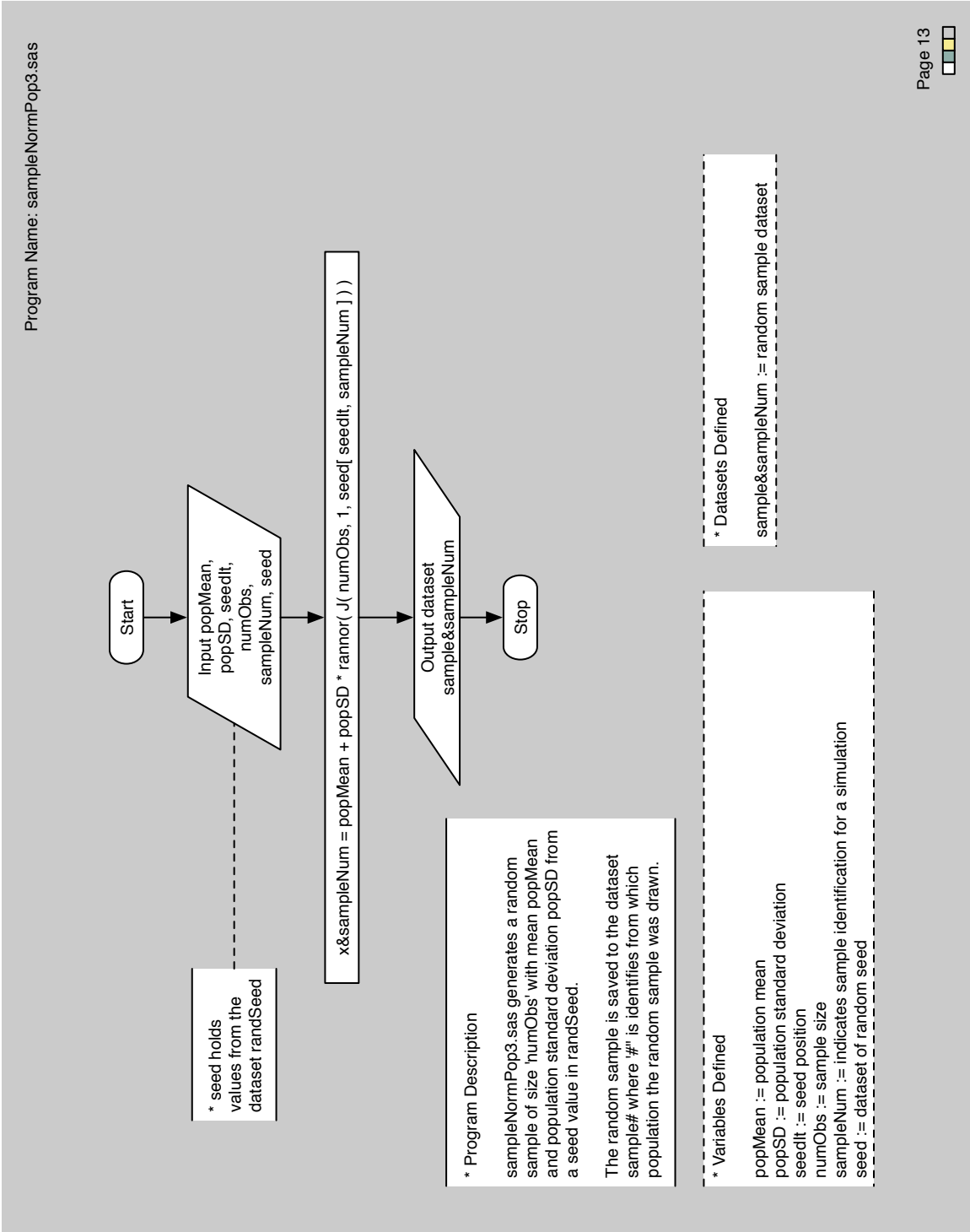
ndf1 := numerator degrees of freedom  
ddf2 := denominator degrees of freedom  
ncp := noncentrality parameter  
slAlpha := level of significance  
tol := tolerance for Golden Section Search method  
alphaA := lower bound for optimized lower alpha  
alphaB := upper bound for optimized lower alpha  
c := Golden ratio constant reduction factor  
x1 := percentile associated with alphaA  
wx1 := height associated with x1  
x2 := percentile associated with alphaB  
wx2 := height associated with x2



Appendix A: Three-Sample Flow Chart of Power Simulations

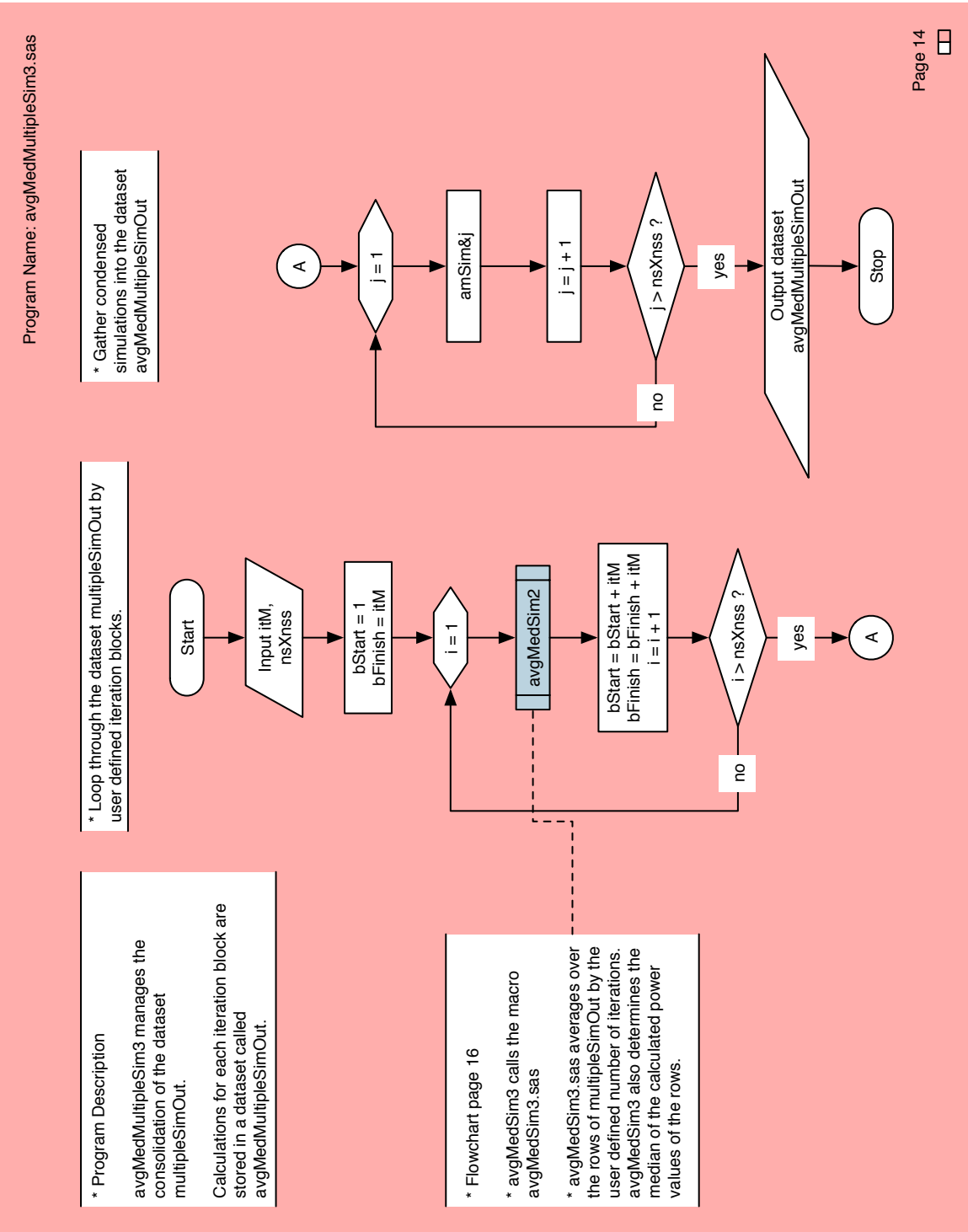


Appendix A: Three-Sample Flow Chart of Power Simulations

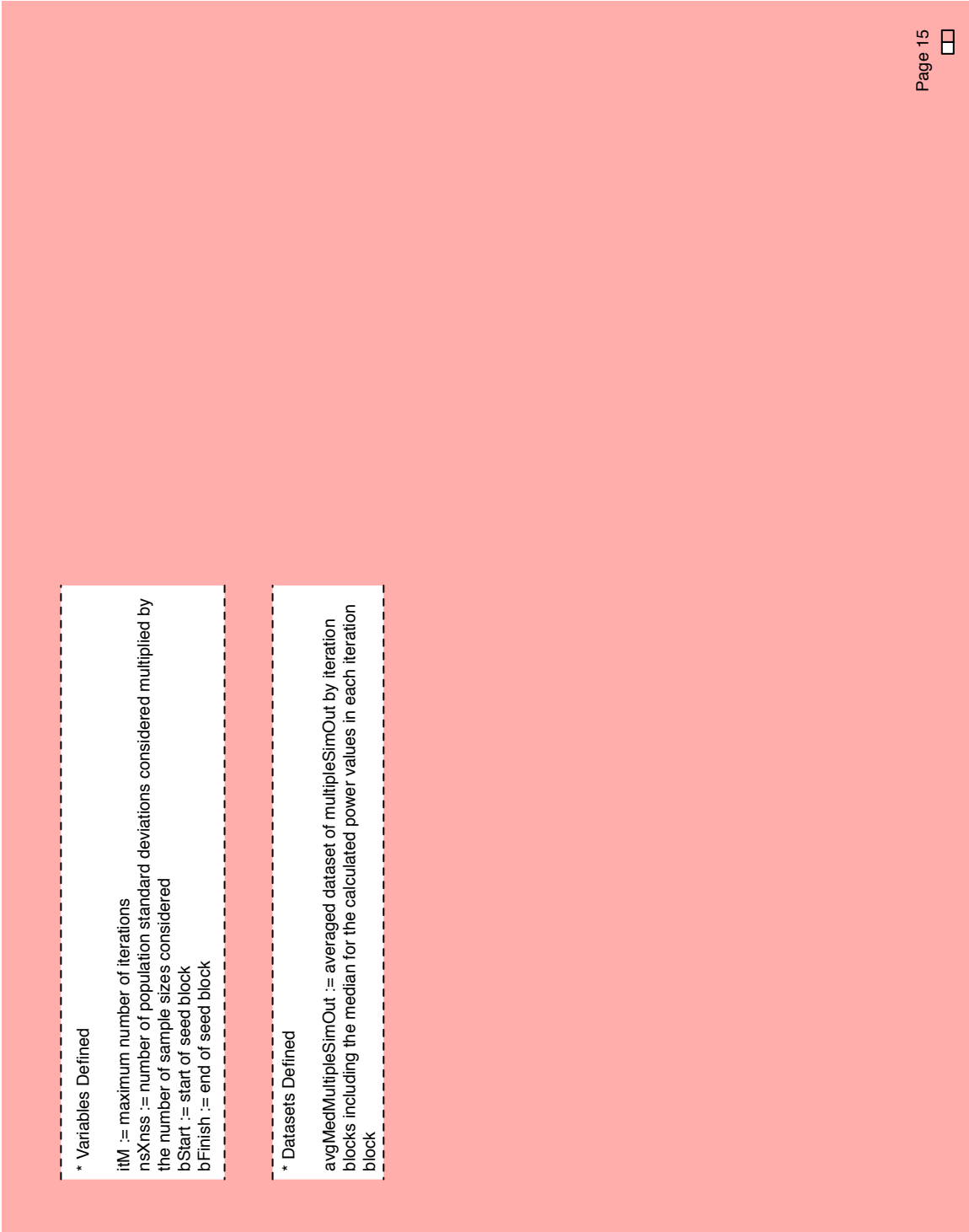




Appendix A: Three-Sample Flow Chart of Power Simulations



Appendix A: Three-Sample Flow Chart of Power Simulations



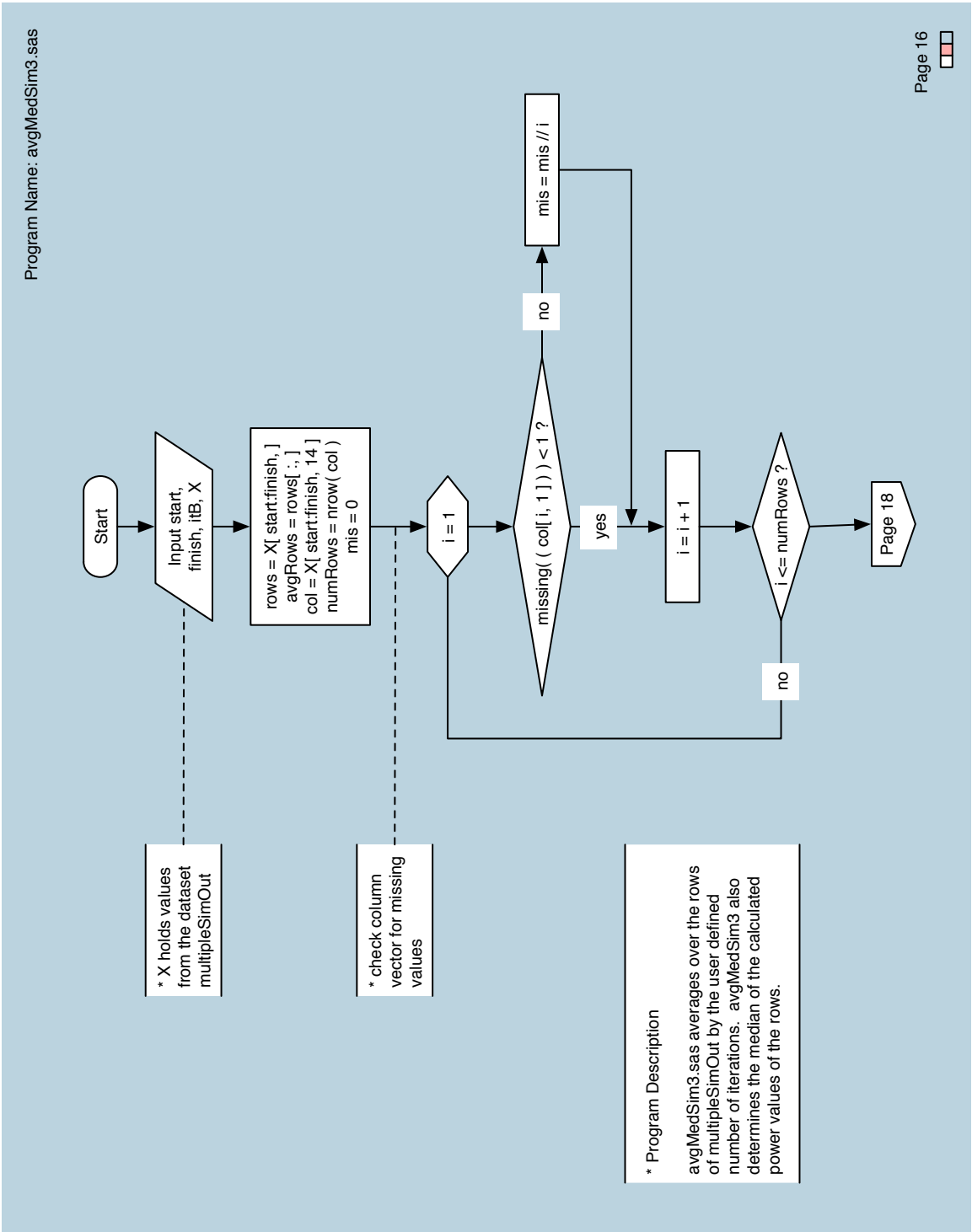
\* Variables Defined

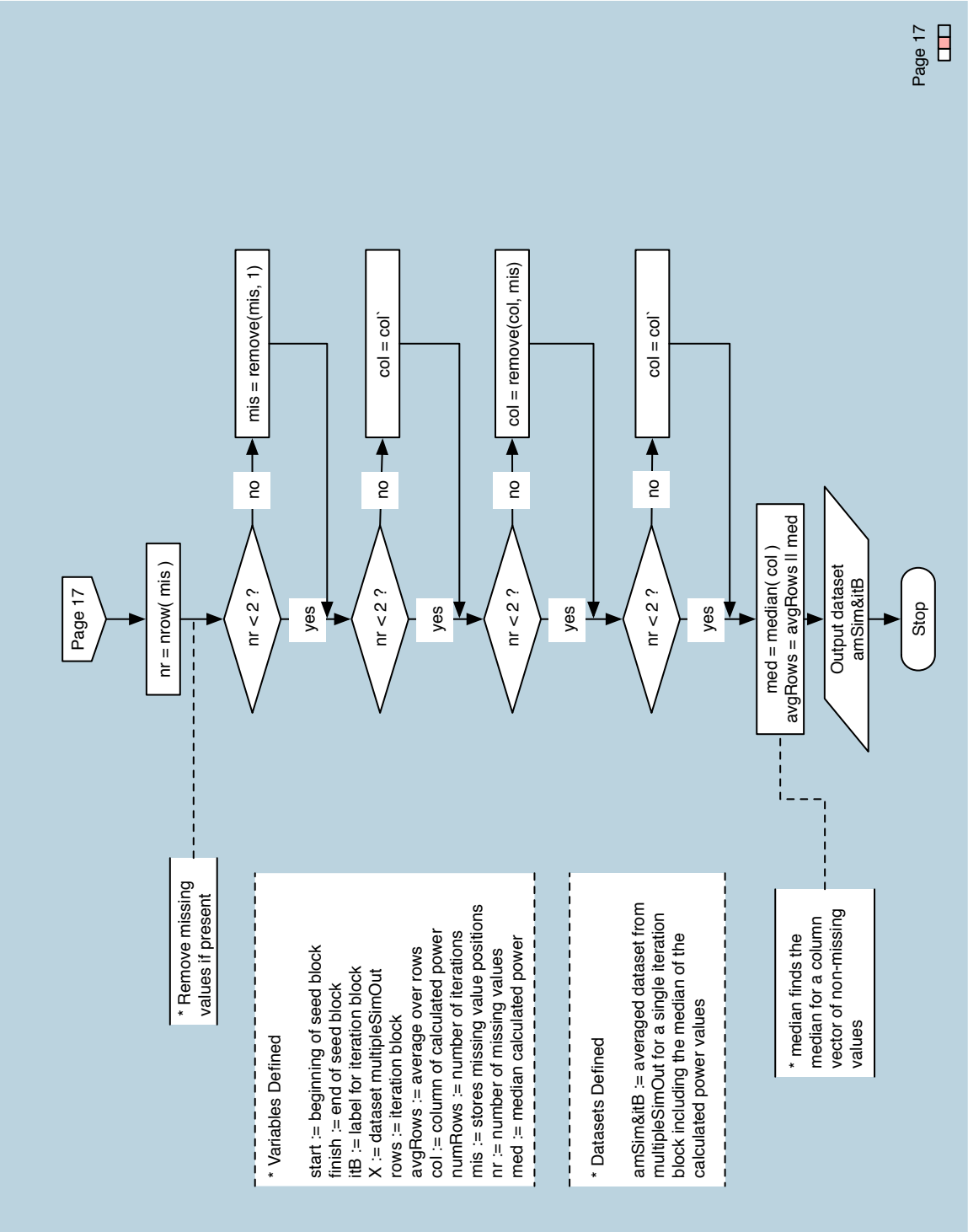
itM := maximum number of iterations  
nsXnss := number of population standard deviations considered multiplied by  
the number of sample sizes considered  
bStart := start of seed block  
bFinish := end of seed block

\* Datasets Defined

avgMedMultipleSimOut := averaged dataset of multipleSimOut by iteration  
blocks including the median for the calculated power values in each iteration  
block

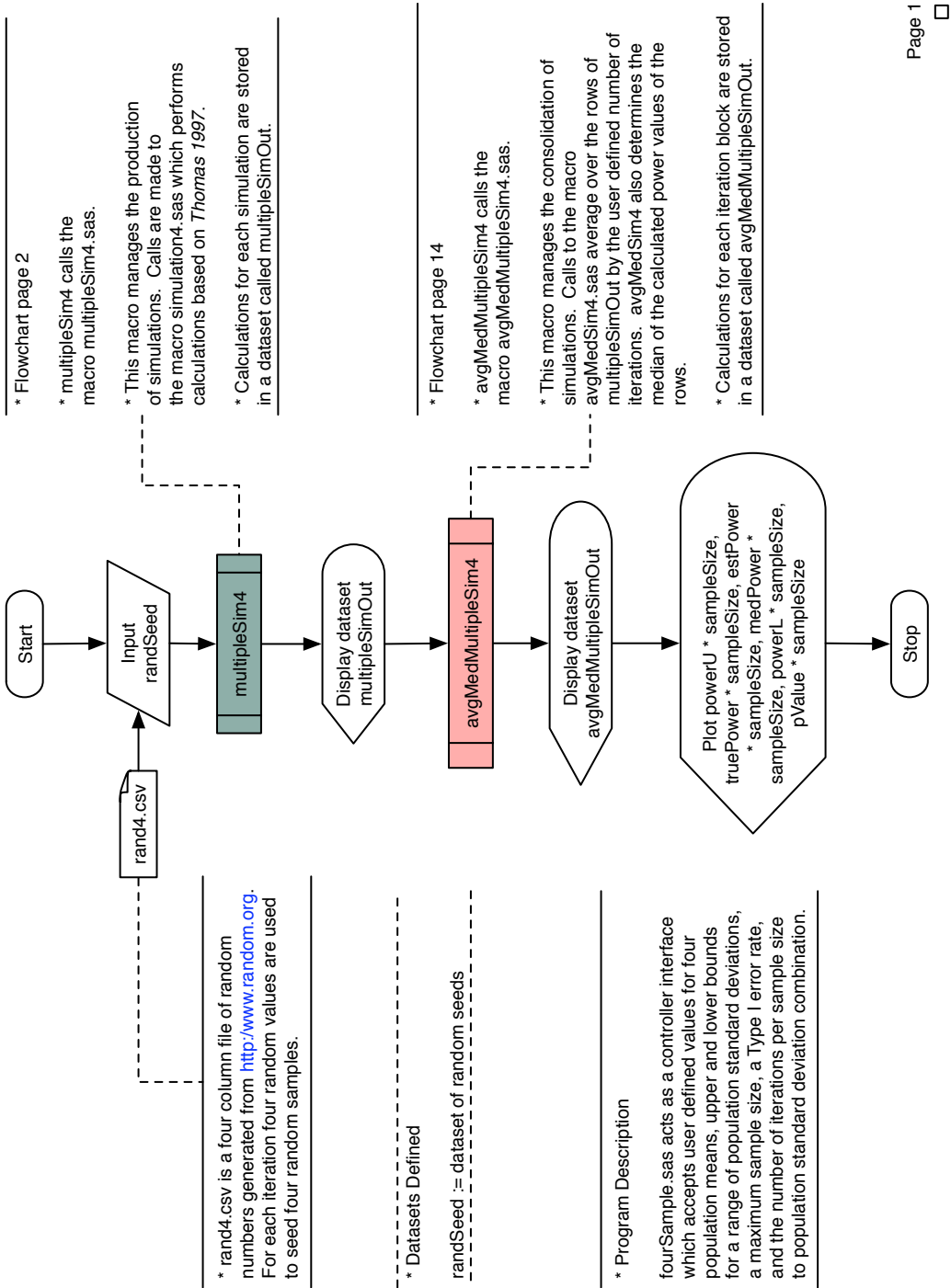
Appendix A: Three-Sample Flow Chart of Power Simulations



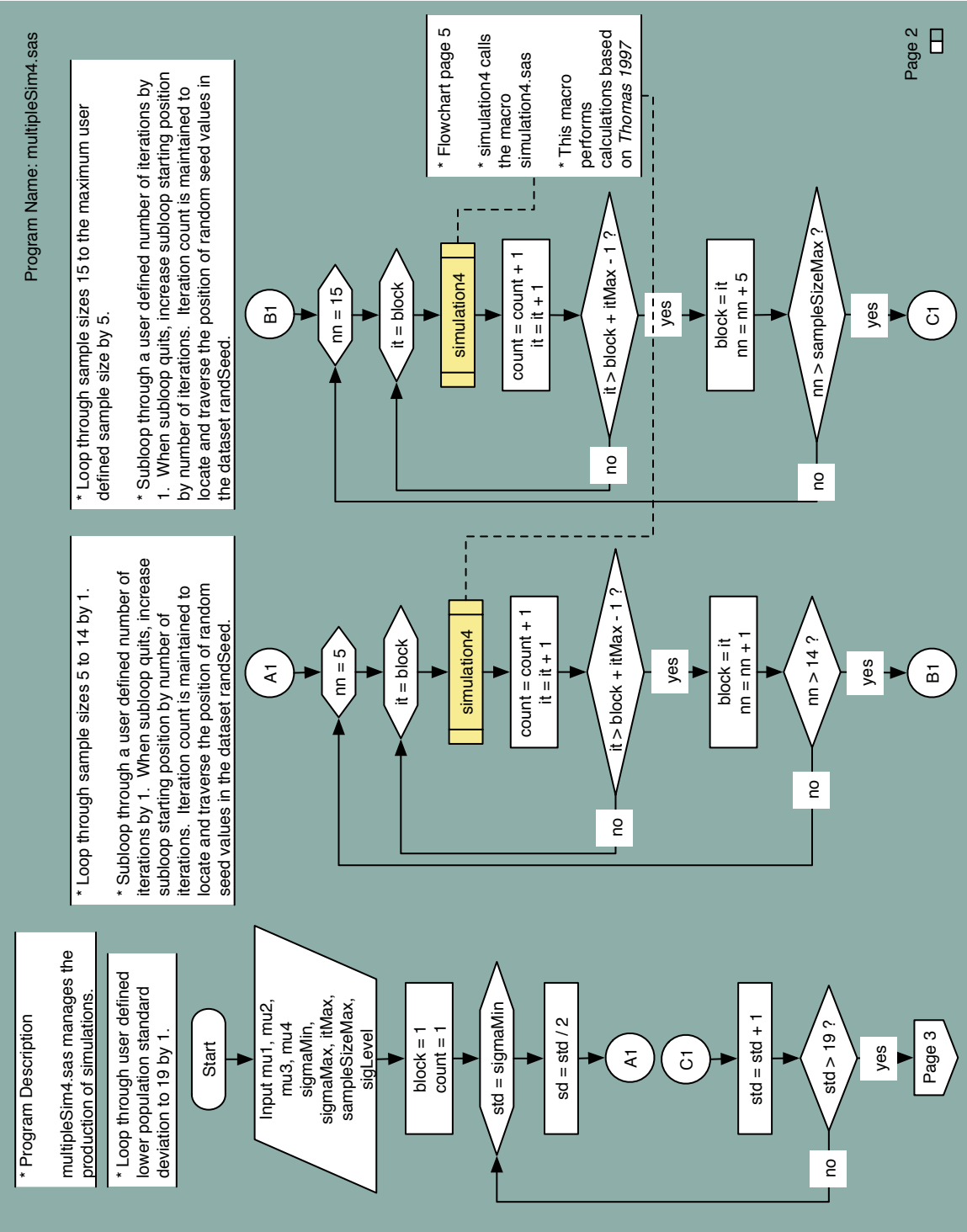


Appendix A: Four-Sample Flow Chart of Power Simulations

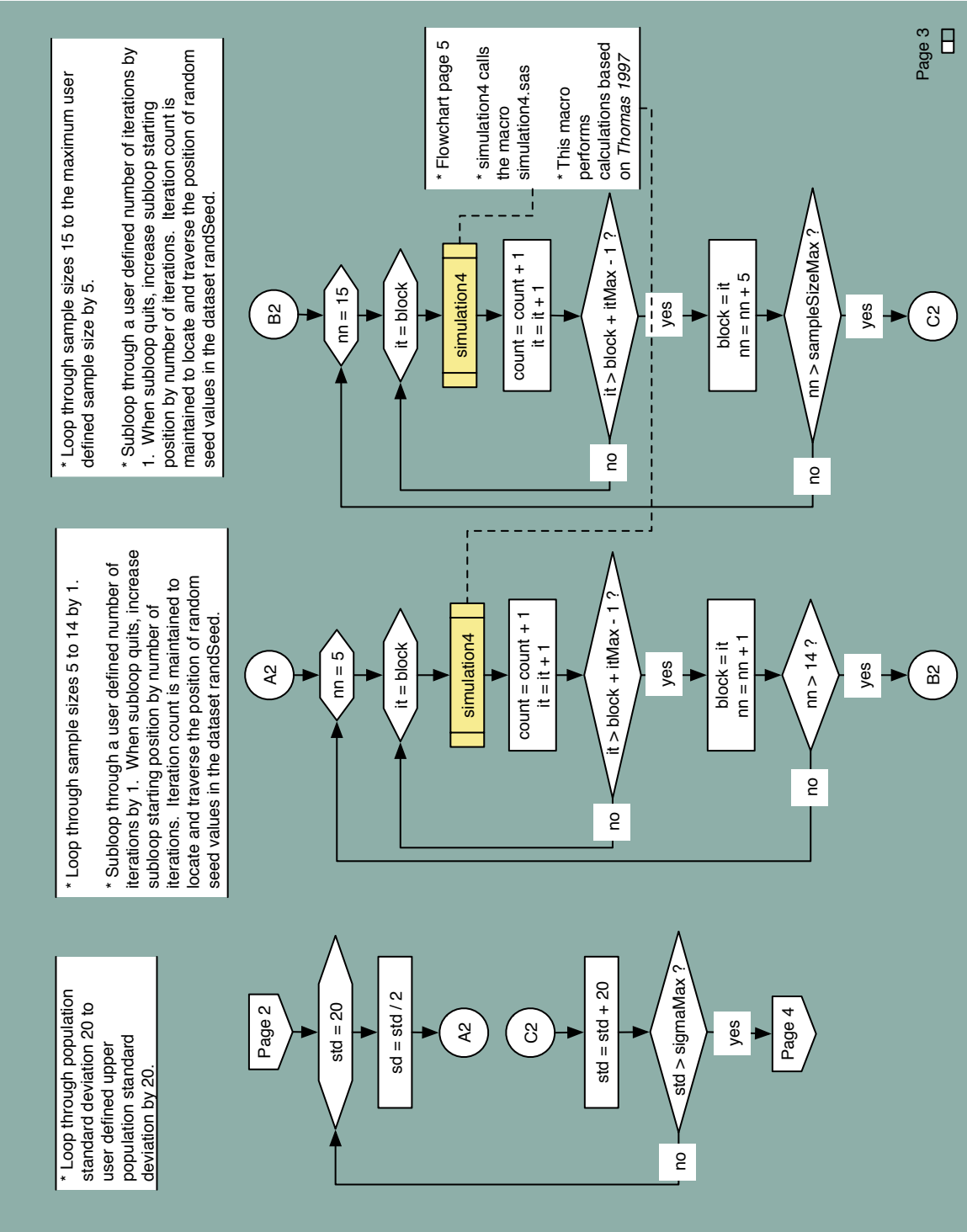
Program Name: fourSample.sas



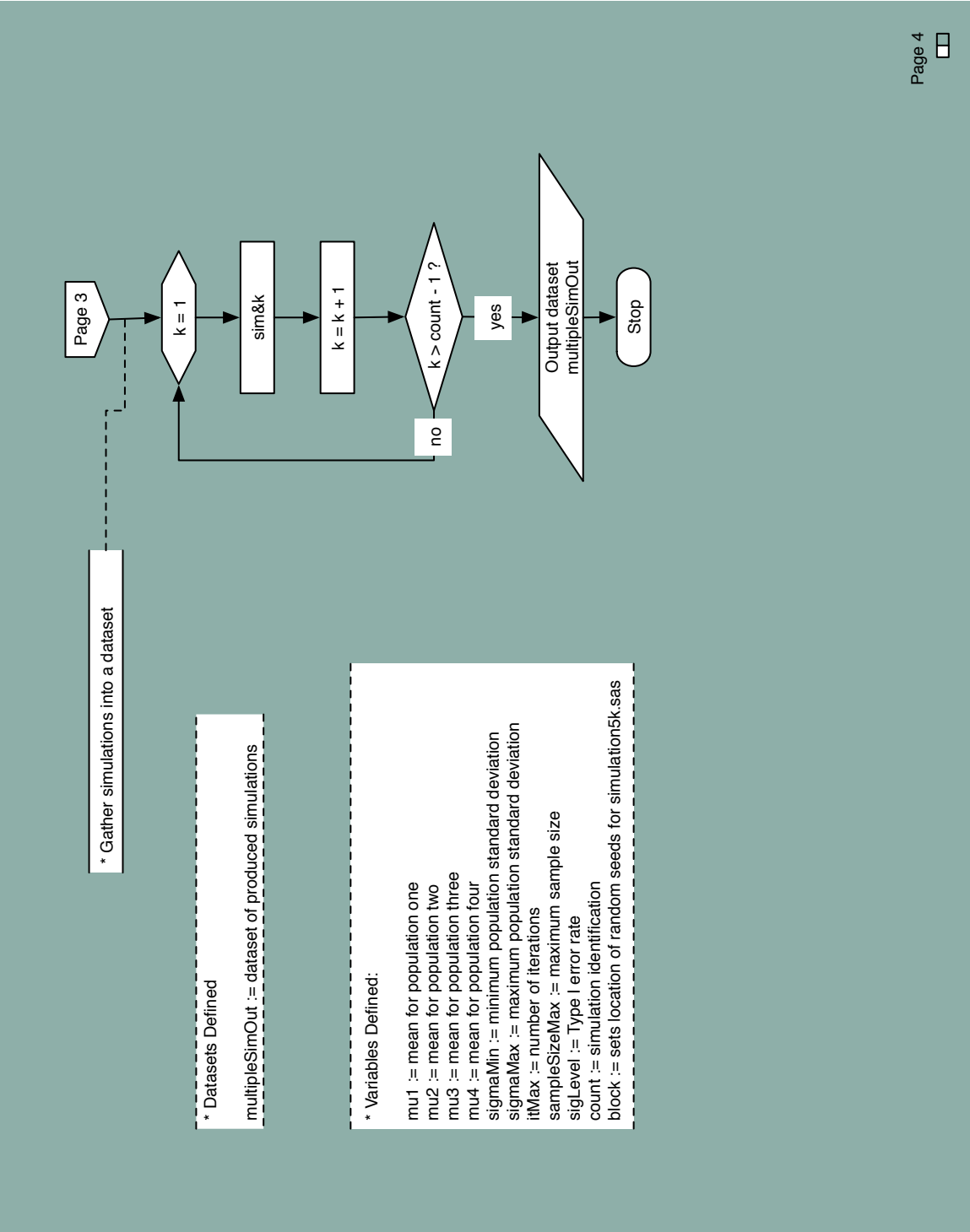
Appendix A: Four-Sample Flow Chart of Power Simulations



Appendix A: Four-Sample Flow Chart of Power Simulations

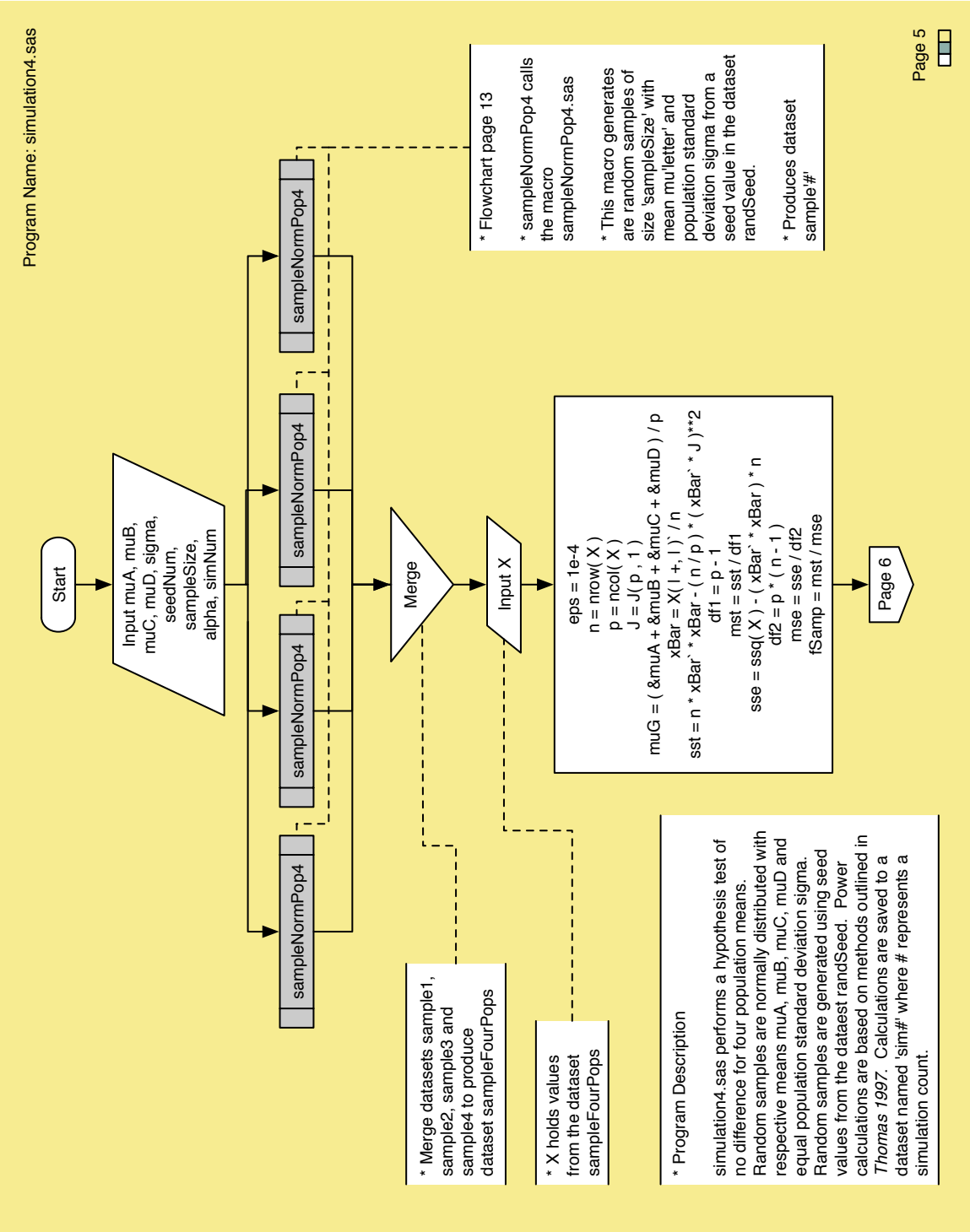


Appendix A: Four-Sample Flow Chart of Power Simulations

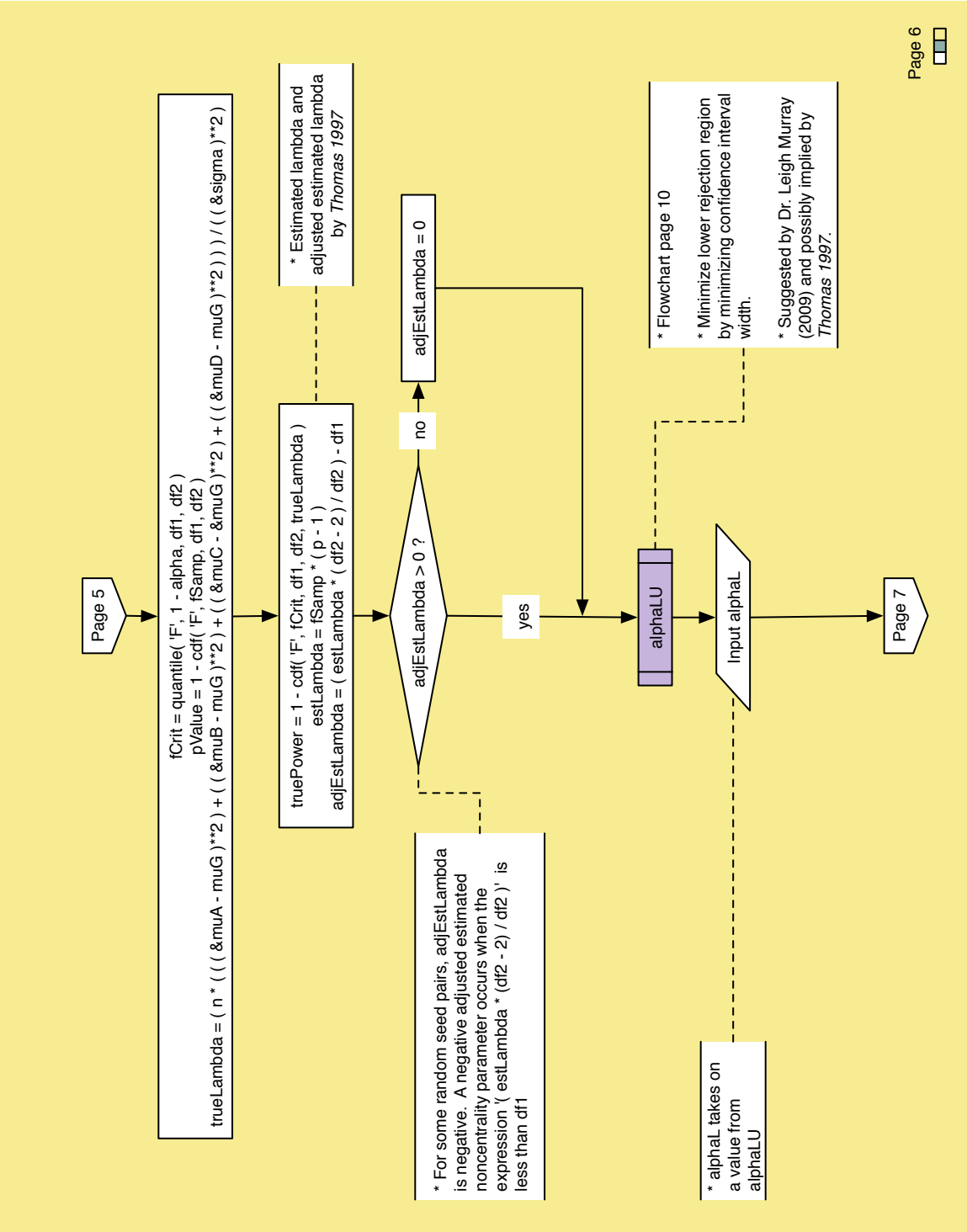




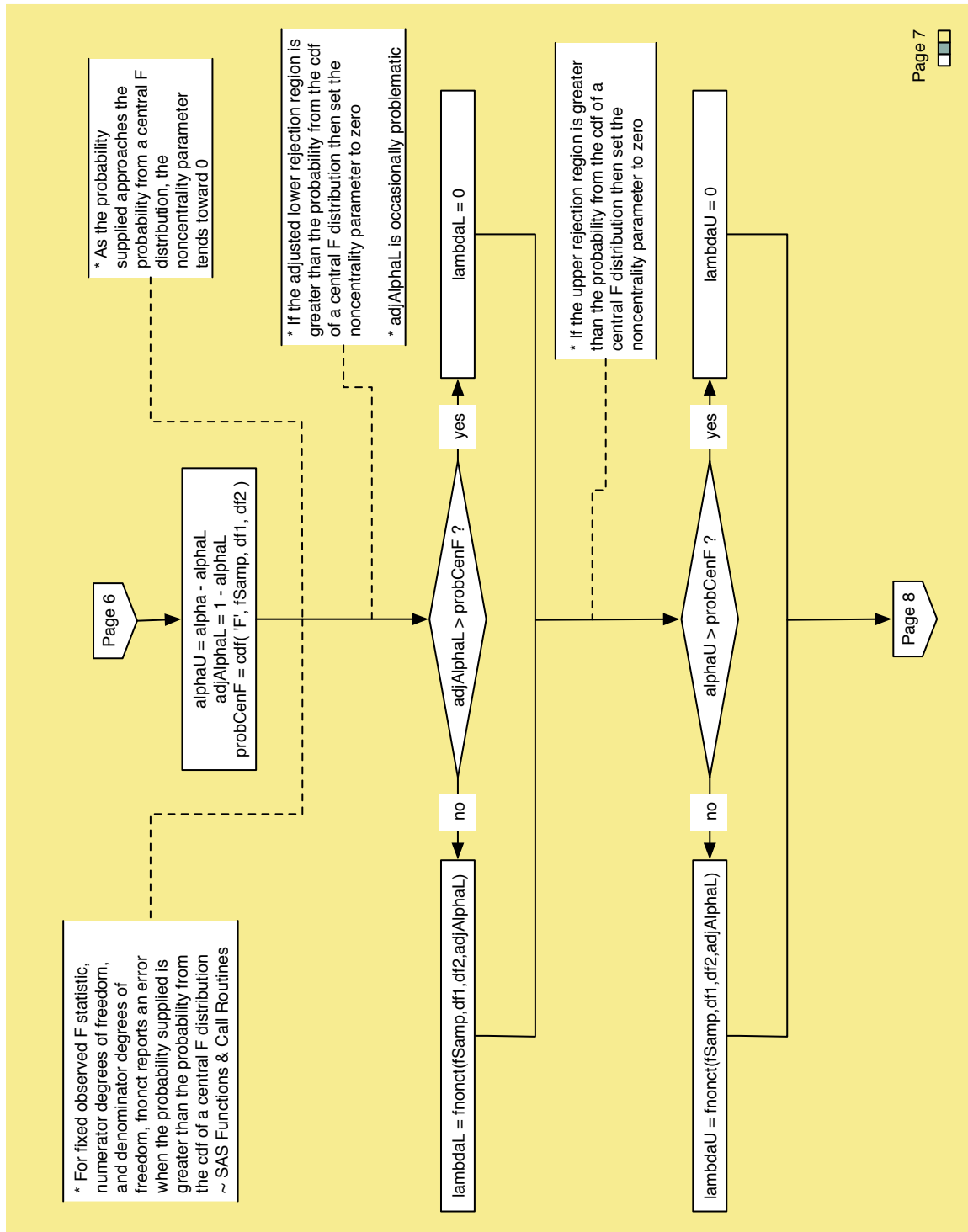
Appendix A: Four-Sample Flow Chart of Power Simulations



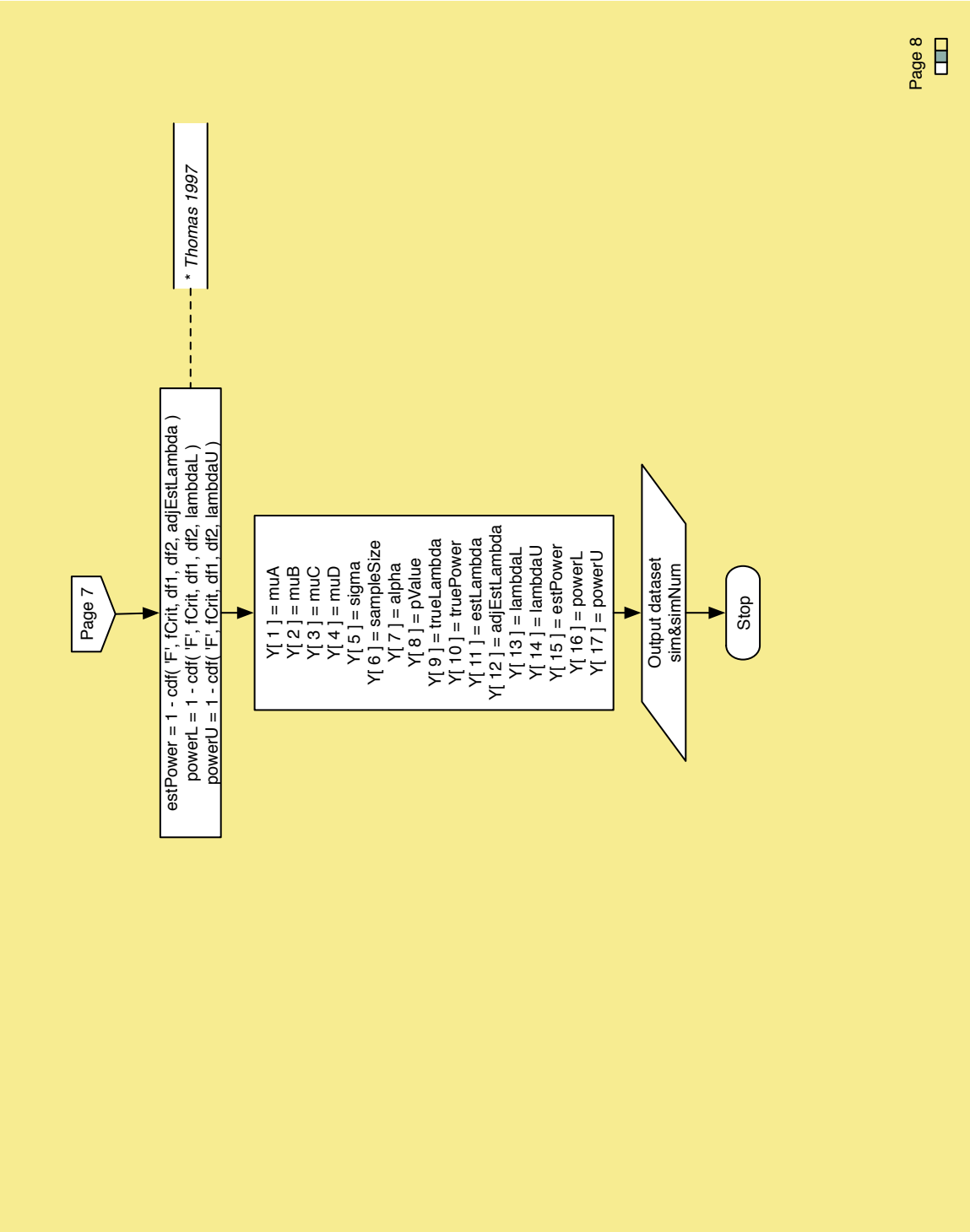
Appendix A: Four-Sample Flow Chart of Power Simulations



## Appendix A: Four-Sample Flow Chart of Power Simulations



Appendix A: Four-Sample Flow Chart of Power Simulations

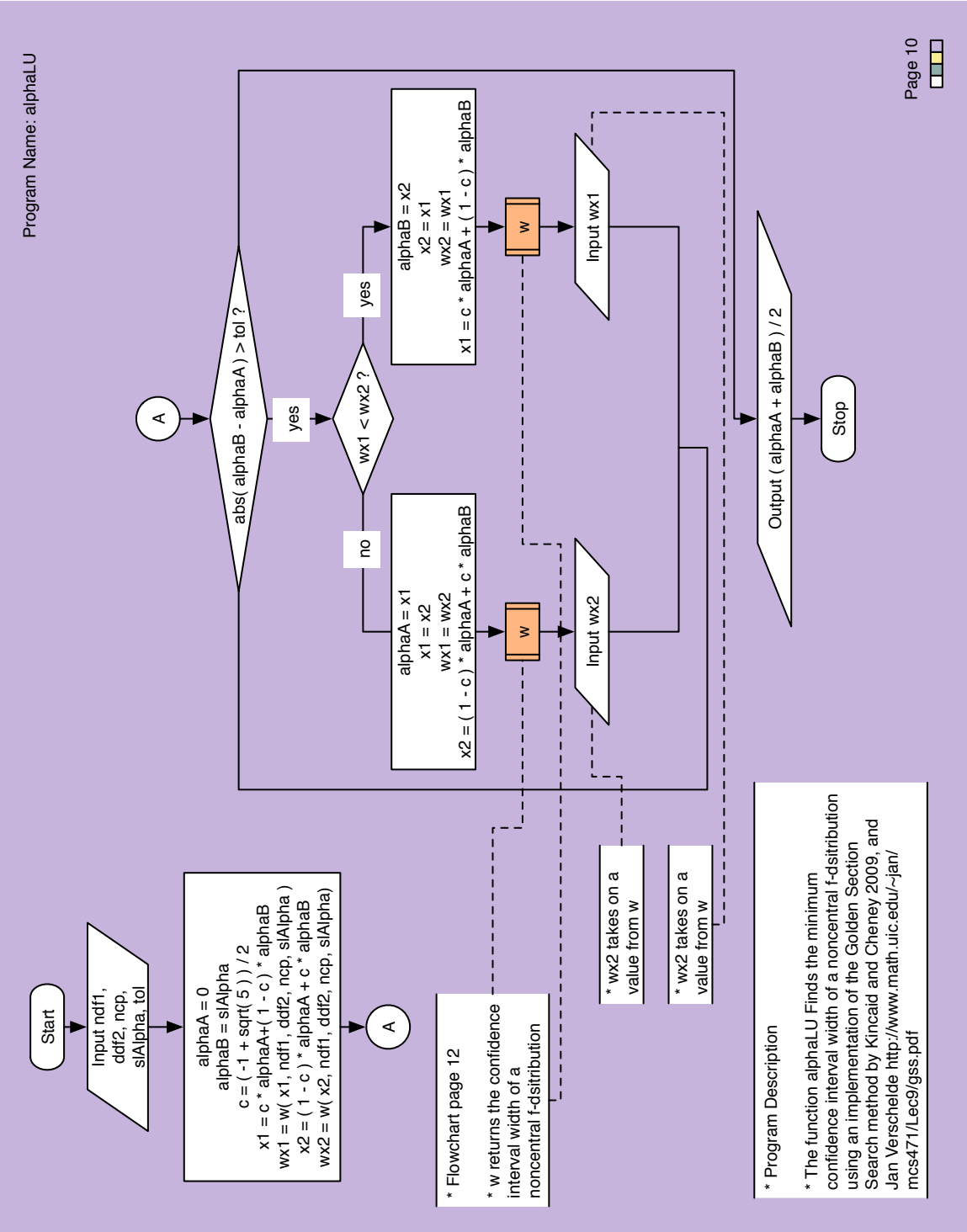


Appendix A: Four-Sample Flow Chart of Power Simulations

* Variables Defined	* Datasets Defined
muA := mean for population one	sampleFourPops := random samples dataset
muB := mean for population two	sim&simNum := simulation dataset
muC := mean for population three	
muD := mean for population four	
muG := average of the tree population means	
sigma := population standard deviation	
seedNum := location of random seeds	
sampleSize := sample size	
alpha := level of significance	
simNum := simulation identification	
X := samples from normal populations	
eps := tolerance for Golden Section Search methods	
n := number rows	
p := number columns	
xBar := sample mean	
sst := sum of squares for treatment	
df1 := numerator degrees of freedom	
mst := mean square for treatment	
sse := sum of squares for error	
df2 := denominator degrees of freedom	
mse := mean square for error	
fSamp := observed F statistic	
fCrit := critical value from a central F distribution	
pValue := p-value for the test	
truePower := noncentrality parameter	
estPower := power for the test	
estLambda := calculated noncentrality parameter	
adjEstLambda := adjusted calculated noncentrality parameter	
alphaL := optimized lower critical region	
alphaU := optimized upper critical region	
probCenF := probability of observed F statistic of the central F distribution	
estPower := calculated power	
powerL := lower confidence interval for calculated power	
powerU := upper confidence interval for calculated power	



Appendix A: Four-Sample Flow Chart of Power Simulations

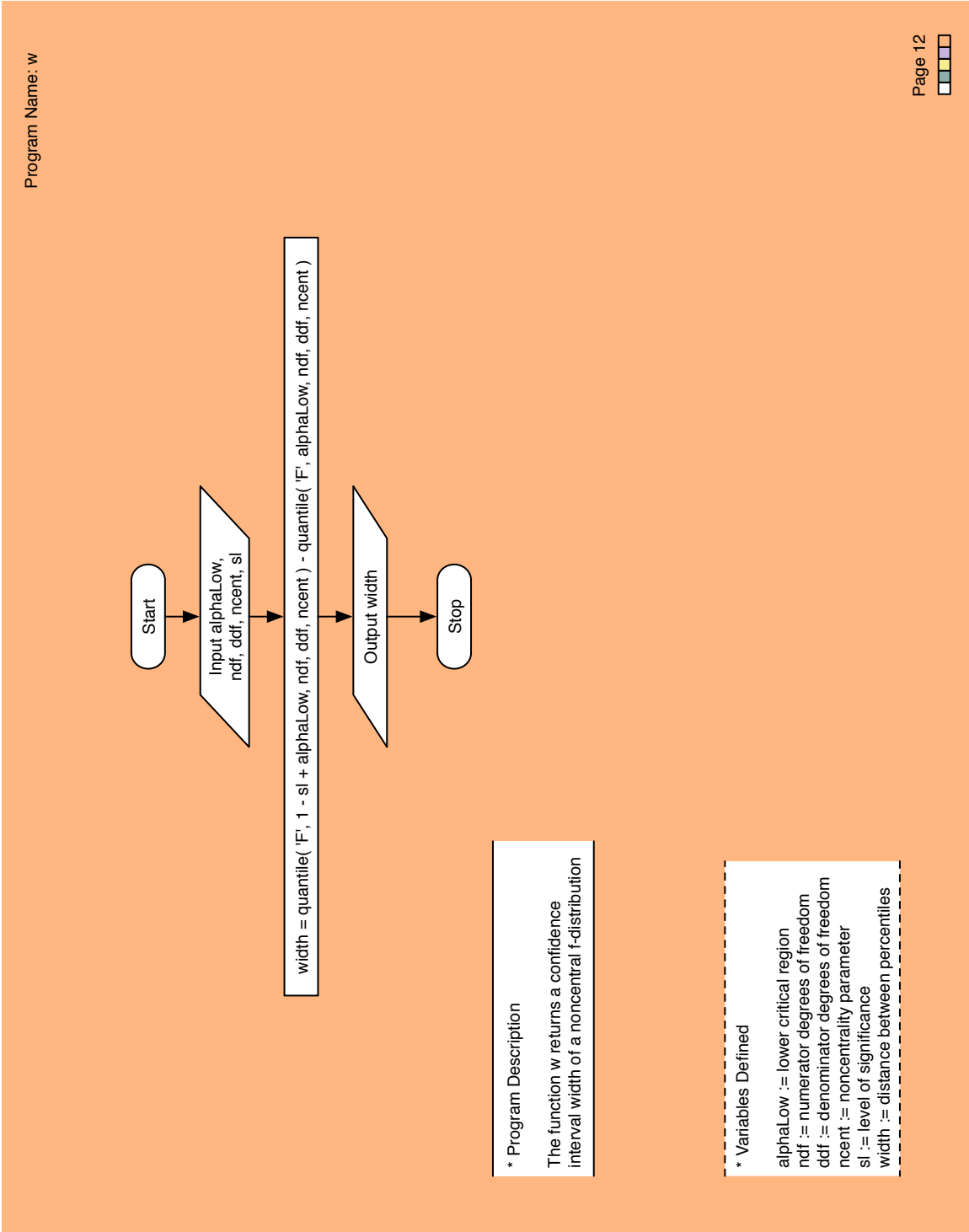


\* Variables Defined

ndf1 := numerator degrees of freedom  
ddf2 := denominator degrees of freedom  
ncp := noncentrality parameter  
slAlpha := level of significance  
tol := tolerance for Golden Section Search method  
alphaA := lower bound for optimized lower alpha  
alphaB := upper bound for optimized lower alpha  
c := Golden ratio constant reduction factor  
x1 := percentile associated with alphaA  
wx1 := height associated with x1  
x2 := percentile associated with alphaB  
wx2 := height associated with x2

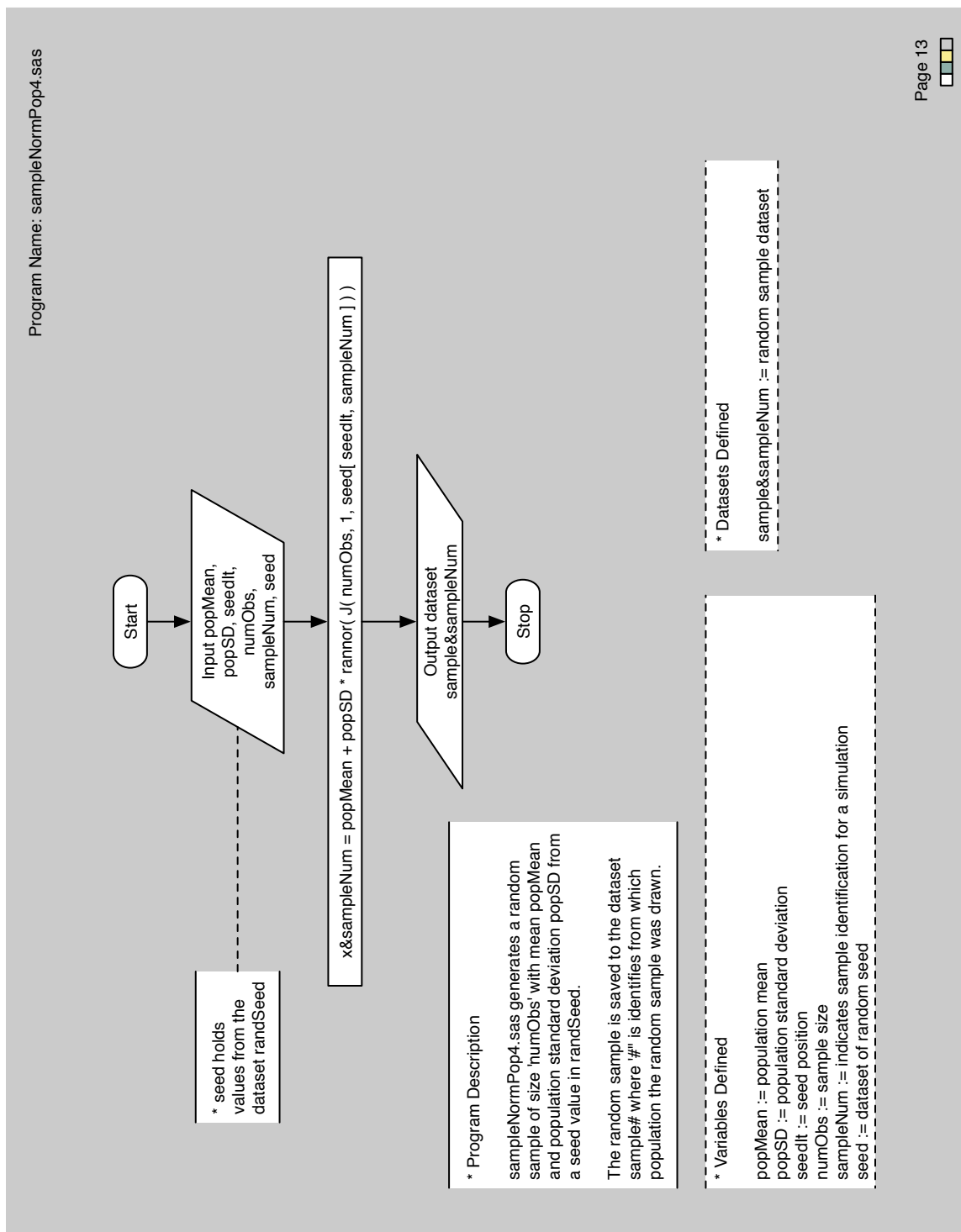


Appendix A: Four-Sample Flow Chart of Power Simulations

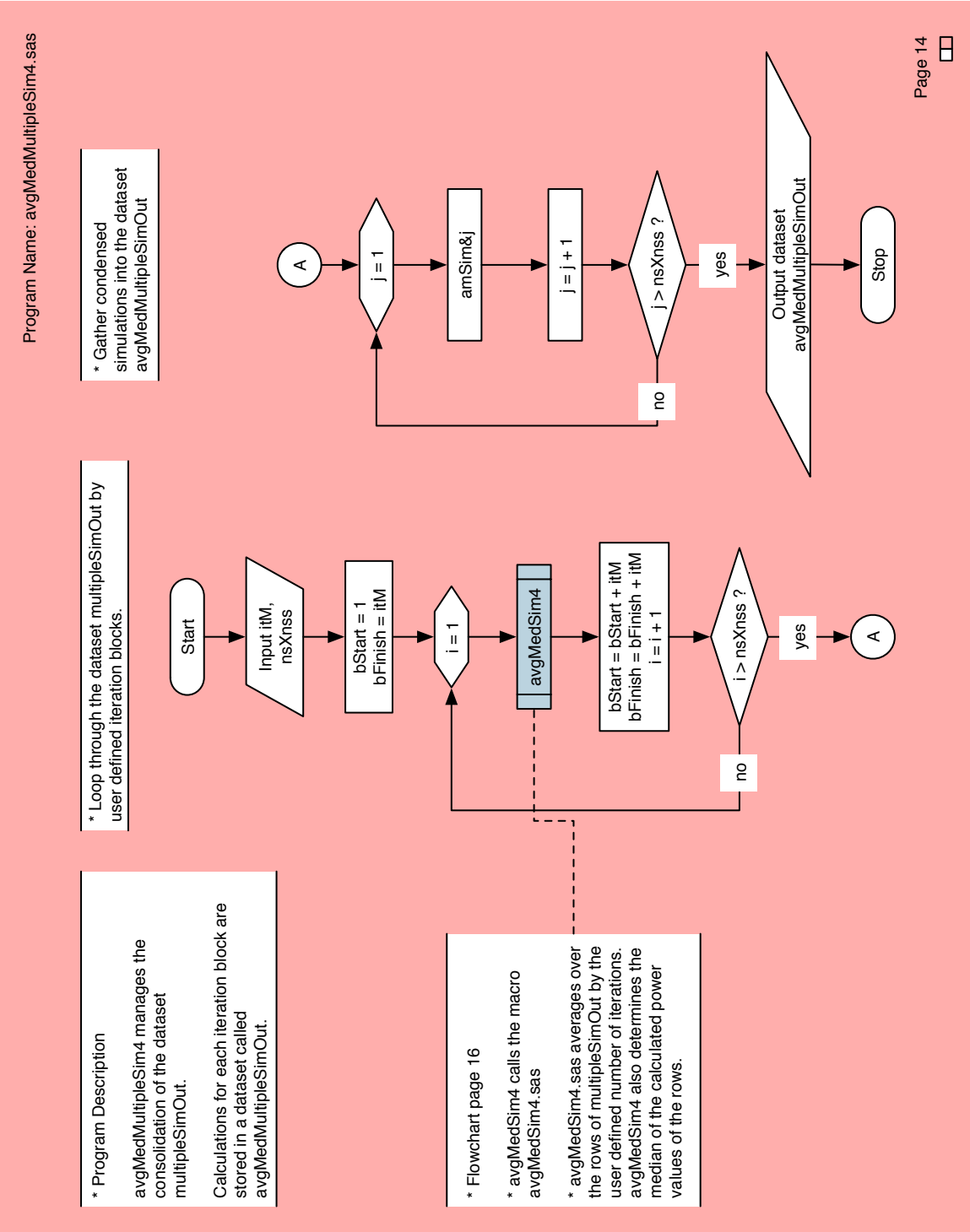




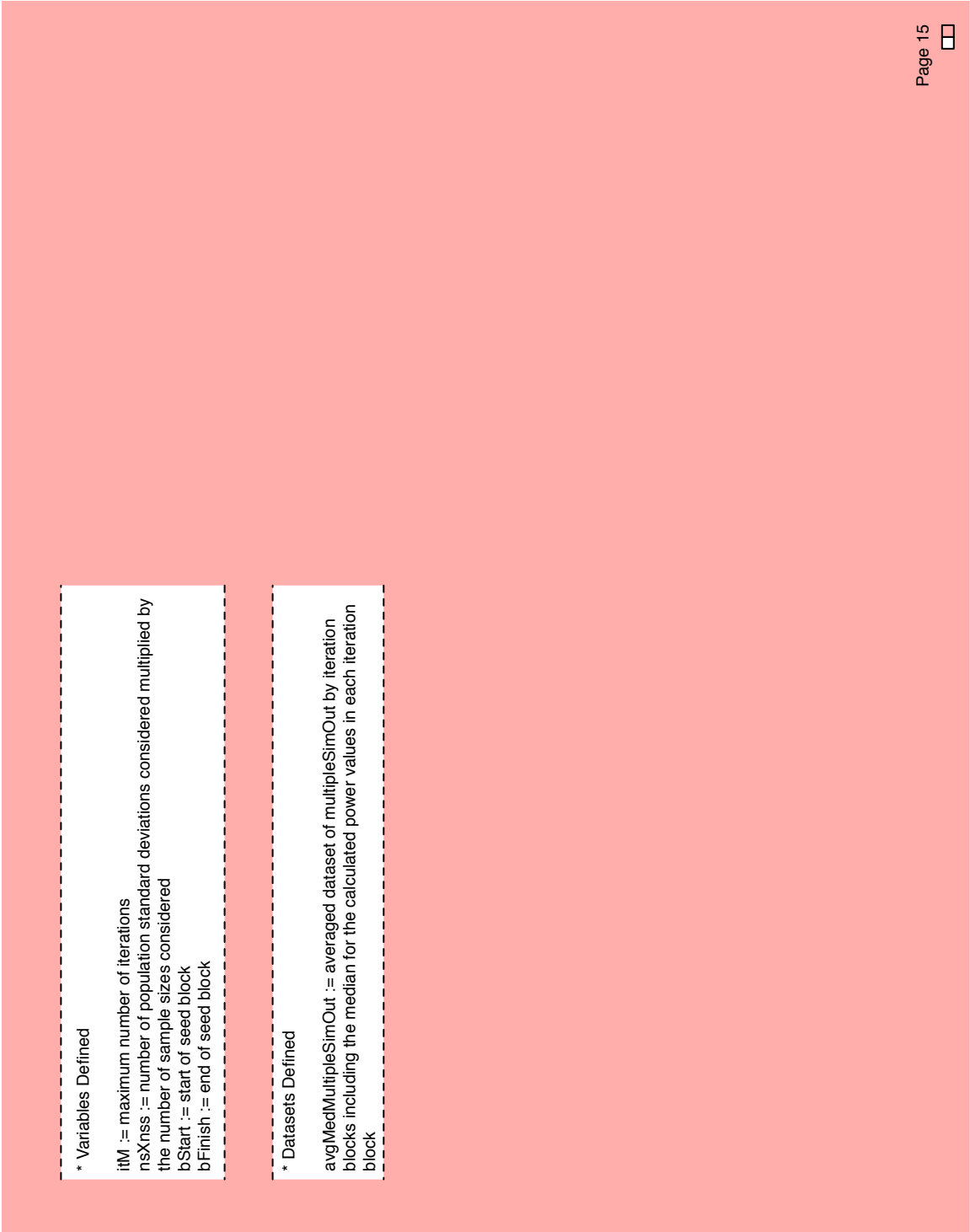
## Appendix A: Four-Sample Flow Chart of Power Simulations



Appendix A: Four-Sample Flow Chart of Power Simulations



Appendix A: Four-Sample Flow Chart of Power Simulations



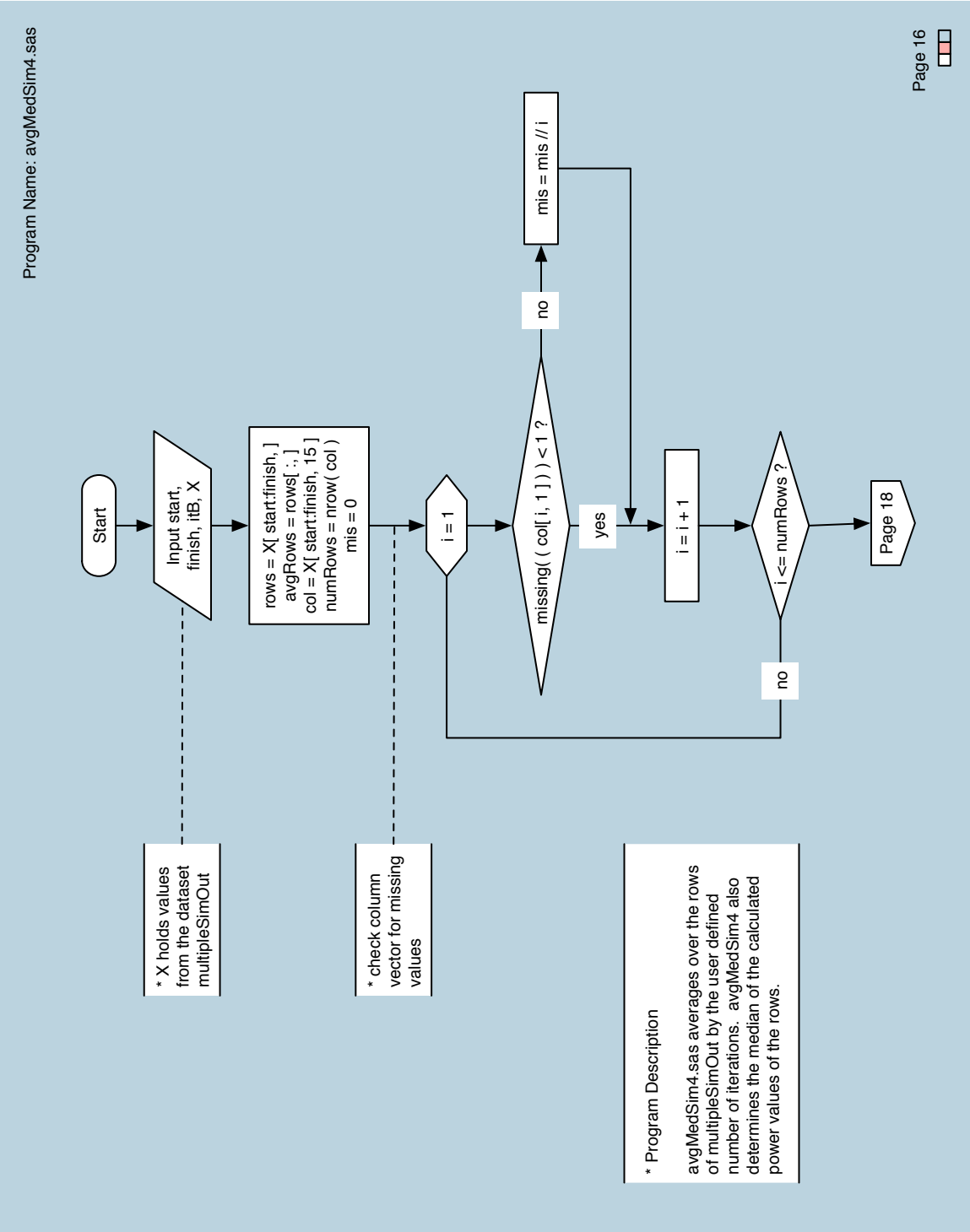
\* Variables Defined

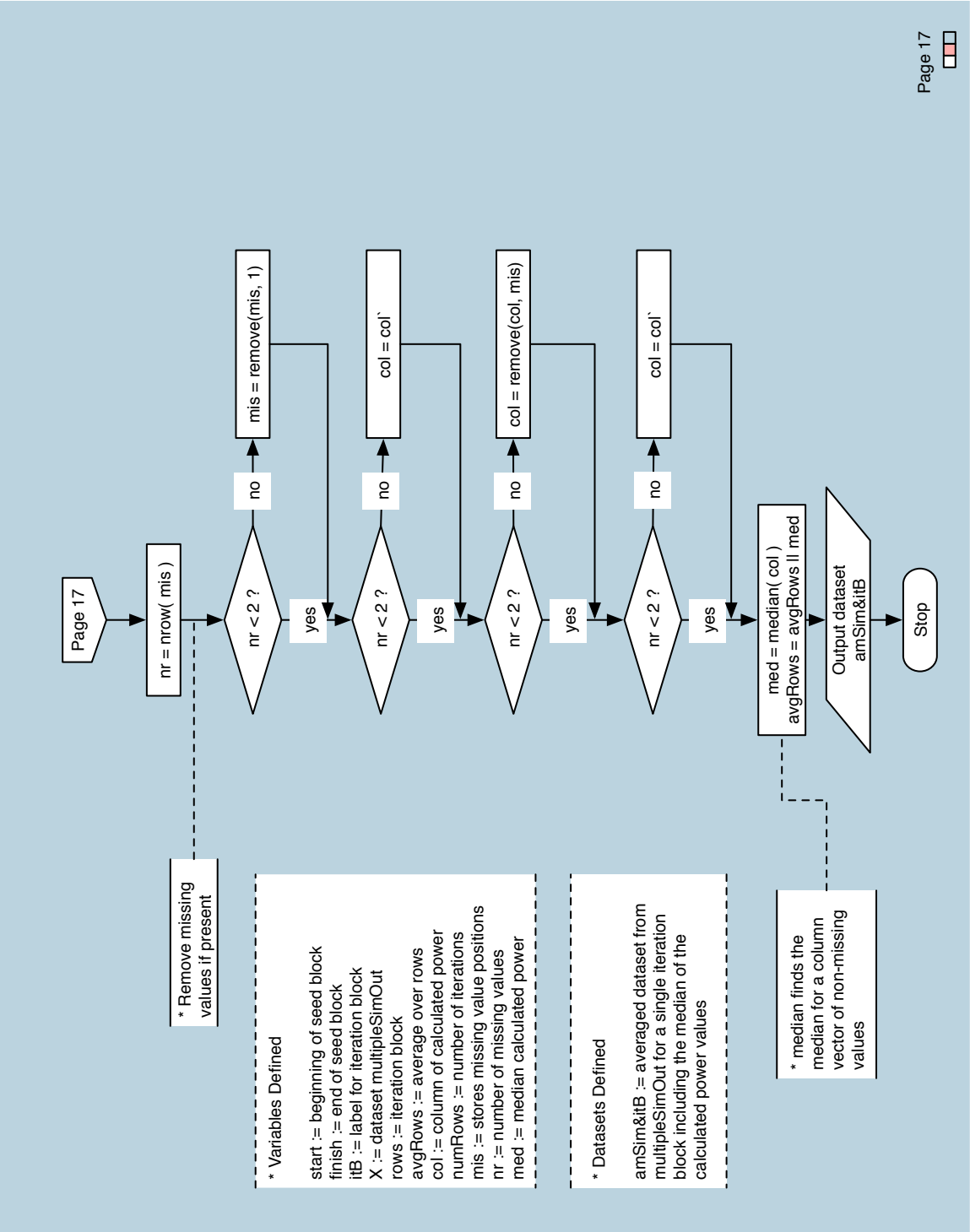
itM := maximum number of iterations  
nsXnss := number of population standard deviations considered multiplied by  
the number of sample sizes considered  
bStart := start of seed block  
bFinish := end of seed block

\* Datasets Defined

avgMedMultipleSimOut := averaged dataset of multipleSimOut by iteration  
blocks including the median for the calculated power values in each iteration  
block

Appendix A: Four-Sample Flow Chart of Power Simulations





## Appendix B: Two-Sample SAS Code for Power Simulations

```

1 /* twoSample ***** */
2 /* */
3 /* Requirement(s): */
4 /* rand2.csv, multiplesim2.sas, avgMedMultiplesim2.sas */
5 /* */
6 /* Program Description: */
7 /* twoSample.sas acts as a controller interface which accepts user */
8 /* defined values for two population means, upper and lower bounds */
9 /* for a range of population standard deviations, a maximum sample size, */
10 /* a type I error rate, and the number of iterations per sample size */
11 /* to population standard deviation combination. */
12 /* */
13 /****** */
14
15
16 proc printto print='\Path\to\Output\Folder\twoSampleOut.lst' log='\Path\to\Output\Folder\twoSampleOut.log';
17 run;
18
19
20 /* Input random seed file to the dataset randseed ***** */
21 /* */
22 /* rand2.csv is a two column file of random numbers generated from */
23 /* http://www.random.org. For each iteration of multiplesim2 two random */
24 /* values are used to seed two random samples. */
25 /* */
26 /****** */
27
28 data randseed;
29   infile "\Path\to\File\rand2.csv" dlm=' ';
30   input seedCol1 seedCol2;
31 run;
32
33
34 /* Produce simulations ***** */
35 /* */
36 /* The multiplesim2 manages the production of simulations. */
37 /* Calls are made to the macro simulation2.sas which performs */
38 /* calculations based on Thomas 1997. */
39 /* */
40 /* Calculations for each simulation are stored in a dataset called */
41 /* multiplesimOut. */
42 /* */
43 /****** */
44
45 %multiplesim2(mu1=30,mu2=40,sigmaMin=5,sigmaMax=40,itMax=100,sampleSizeMax=50,sigLevel=0.05);
46
47

```

## Appendix B: Two-Sample SAS Code for Power Simulations

```

48 /* Print multiplesimOut */
49
50 proc print data=multiplesimOut;
51 run;
52
53
54 /* Produce averages and medians for simulations ***** */
55 /*
56 /* The avgMedMultipleSim2 manages the consolidation of simulations.
57 /* Calls are made to avgMedSim2 for each sigma by sample size
58 /* combination. avgMedSim2 averages over a user specified number
59 /* of rows from multiplesimOut. avgMedSim2 also determines the
60 /* median of the calculated power values of the rows.
61 /*
62 /* Calculations for each iteration block are stored in a dataset
63 /* called avgMedMultipleSimOut.
64 /*
65 /******
66
67 %avgMedMultipleSim2(itM=100,nsXnss=306);
68
69
70 /* Print avgMedMultipleSimOut */
71
72 proc print data=avgMedMultipleSimOut;
73 run;
74
75
76 /* Plot avgMedMultipleSimOut */
77
78 goptions ftitle=swiss ftext=swiss;
79 symbol1 value=square interpol=join width=0.5 color=blue;
80 symbol2 value=star interpol=join width=0.5 color=red;
81 symbol3 value=triangle interpol=join width=0.5 color=yellow;
82 symbol4 value='x' interpol=join width=0.5 color=orange;
83 symbol5 value=circle interpol=join width=0.5 color=blue;
84 symbol6 value=dot interpol=join width=0.5 color=green;
85 title2 height=1.4 "Sample Size vs Power";
86 axis1 order=(0.0 to 1.0 by 0.1) label=(angle=90 'Power') minor=none;
87 axis2 order=(5 to 50 by 5) label=(angle=0 'Sample Size');
88 legend label=none value=(h=.8 'Upper AVG Estimated Power' 'True Power' 'AVG Estimated Power' 'Median Power' 'Lower
... AVG Estimated Power' 'AVG P-Value')
89 position=center;
90
91
92
93

```

## Appendix B: Two-Sample SAS Code for Power Simulations

```
94 proc gplot data=avgMedMultipleSimOut;  
95   plot powerU*sampleSize truePower*sampleSize estPower*sampleSize medPower*sampleSize powerL*sampleSize  
   pValue*sampleSize/overlay vaxis=axis1 haxis=axis2 hminor=4 legend=legend1;  
96   by sigma;  
97 run;
```

---



## Appendix B: Two-Sample SAS Code for Power Simulations

```

1 /* multiplesim2 ***** */
2 /* */
3 /* Requirement(s): */
4 /* simulation2.sas */
5 /* */
6 /* Program Description: */
7 /* multiplesim2 Iterates simulation2.sas and combines simulation output to a */
8 /* dataset. */
9 /* */
10 /* Input: */
11 /* mu1          := population one mean */
12 /* mu2          := population two mean */
13 /* sigmaMin     := minimum population standard deviation controller */
14 /* sigmaMax     := maximum population standard deviation controller */
15 /* itMax        := number of iterations */
16 /* sampleSizeMax := largest sample size considered */
17 /* sigLevel     := level of significance */
18 /* */
19 /* Note(s): */
20 /* In the subsubloop iterations run in blocks. This is done to */
21 /* traverse the dataset randSeed, where by each call to the macro */
22 /* simulation2 gets a different random seed position. */
23 /* */
24 /* This macro contains two main loops. The first main loop */
25 /* iterates from the minimum user define population standard deviation to 19 */
26 /* by 1. The second main loop iterates from 20 to the maximum user defined */
27 /* population standard deviation by 20. Each main loop has two subloops. */
28 /* The first subloop iterates from a system defined minimum sample size of 5 */
29 /* to 14 by 1. The second subloop iterates from 15 to the maximum user */
30 /* defined sample size by 5. Each subloop has a subsubloop. */
31 /* Each subsubloop iterates by a use defined number of iterations. After */
32 /* a subsubloop executes, the starting position of an iteration 'block' is */
33 /* increased by the the user defined number of iterations. */
34 /* */
35 /****** */
36
37
38 %macro multiplesim2(mu1=,mu2=,sigmaMin=,sigmaMax=,itMax=,sampleSizeMax=,sigLevel=);
39
40 %local std sd nn it block s n i b;
41 %let block = 1;
42 %let b = 1;
43 %let count = 1;
44
45
46
47

```

## Appendix B: Two-Sample SAS Code for Power Simulations

```

48 /* Produce simulations */
49
50 %do std = &sigmaMin %to 19 %by 1;
51   %let sd = %sysevalf(&std/2);
52   %do nn = 5 %to 14 %by 1;
53     %do it = &block %to &block+&itMax-1 %by 1;
54       %simulation2(muA=&mul,muB=&mu2,sigma=&sd,seedNum=&it,sampleSize=&nn,alpha=&sigLevel,simNum=&count);
55       %let count = %eval(&count+1);
56     %end;
57   %let block = &it;
58 %end;
59 %do nn = 15 %to &sampleSizeMax %by 5;
60   %do it = &block %to &block+&itMax-1 %by 1;
61     %simulation2(muA=&mul,muB=&mu2,sigma=&sd,seedNum=&it,sampleSize=&nn,alpha=&sigLevel,simNum=&count);
62     %let count = %eval(&count+1);
63   %end;
64   %let block = &it;
65 %end;
66
67 %do std = 20 %to &sigmaMax %by 20;
68   %let sd = %sysevalf(&std/2);
69   %do nn = 5 %to 14 %by 1;
70     %do it = &block %to &block+&itMax-1 %by 1;
71       %simulation2(muA=&mul,muB=&mu2,sigma=&sd,seedNum=&it,sampleSize=&nn,alpha=&sigLevel,simNum=&count);
72       %let count = %eval(&count+1);
73     %end;
74   %let block = &it;
75 %end;
76 %do nn = 15 %to &sampleSizeMax %by 5;
77   %do it = &block %to &block+&itMax-1 %by 1;
78     %simulation2(muA=&mul,muB=&mu2,sigma=&sd,seedNum=&it,sampleSize=&nn,alpha=&sigLevel,simNum=&count);
79     %let count = %eval(&count+1);
80   %end;
81   %let block = &it;
82 %end;
83
84
85
86
87
88
89
90
91
92
93
94

```

## Appendix B: Two-Sample SAS Code for Power Simulations

```
95 /* Combine simulations into a dataset and name columns */
96
97 data multiplesimOut;
98   set
99     %do k = 1 %to &count-1 %by 1;
100       sim&k
101     %end;
102   ; /* Double do-loop to reference datasets. ';' closes 'set' */
103
104   rename coll1=mu1 coll2=mu2 coll3=sigma coll4=sampleSize coll5=sigLevel coll6=pValue coll7=trueLambda coll8=truePower
105     coll9=estLambda coll10=adjEstLambda coll11=lambdau coll12=estPower coll13=powerL coll14=powerU coll15=powerU;
106
107   run;
108
109 %mend multiplesim2;
```

## Appendix B: Two-Sample SAS Code for Power Simulations

```

1  /* simulation2 ***** */
2  /* */
3  /* Requirement(s): */
4  /* sampleNormPop2.sas */
5  /* */
6  /* Program Description: */
7  /* simulation2.sas performs a hypothesis */
8  /* test of no difference for two population means. Random samples */
9  /* are normally distributed with respective means muA and muB with equal */
10 /* population standard deviation sigma. Random samples are generated */
11 /* using seed values from the dataset randSeed. Power calculations */
12 /* are based on methods outlined in Thomas 1997. Calculations are */
13 /* saved to a dataset named 'sim#' where # represents the */
14 /* simulation count. */
15 /* */
16 /* Input: */
17 /* muA      := mean of population one */
18 /* muB      := mean of population two */
19 /* sigma    := standard deviation of populations one and two */
20 /* seedNum  := position for random seed */
21 /* sampleSize := sample size taken for populations one and two */
22 /* alpha    := level of significance */
23 /* simNum   := simulation ID */
24 /* */
25 /* Note(s): */
26 /* Optimal lower significance level by Golden Section Search method */
27 /* */
28 /****** */
29
30
31 %MACRO simulation2(muA=,muB=,sigma=,seedNum=,sampleSize=,alpha=,simNum=);
32
33     /* Generate random samples from two normal populations */
34
35     %sampleNormPop2(popMean=&muA,popSD=&sigma,seedIt=&seedNum,numObs=&sampleSize,sampleNum=1);
36     %sampleNormPop2(popMean=&muB,popSD=&sigma,seedIt=&seedNum,numObs=&sampleSize,sampleNum=2);
37
38
39     /* Merge datasets sample1 and sample2 to produce dataset sampleTwoPops */
40
41     data sampleTwoPops;
42         merge sample1 sample2;
43     run;
44
45
46
47

```

## Appendix B: Two-Sample SAS Code for Power Simulations

```

48 /* Main */
49
50 proc iml;
51   reset nolog;
52   use sampleTwoPops;
53   read all into X;
54
55   /* Function ***** */
56   /*
57   /*
58   /* Return the confidence interval width for a noncentral
59   /* f-distribution
60   /*
61   /* ***** */
62
63   start w(alphaLow,ndf,ddf,ncent,sl);
64
65     width = quantile('F',1-sl+alphaLow,ndf,ddf,ncent)-quantile('F',alphaLow,ndf,ddf,ncent);
66     return(width);
67
68   finish w;
69
70
71   /* Function: optimal lower critical region ***** */
72   /*
73   /* Finds the minimum confidence interval width using an
74   /* implementation of the Golden Section Search method
75   /* similar to Press et al. (2007).
76   /*
77   /* Note(s):
78   /* The Golden Section Search method finds a value which minimizes
79   /* a function. The function must be one dimensional and unimodal.
80   /*
81   /* Imagine a 'typical' pdf of an F distribution. Say we want to
82   /* setup an interval such that the area covered by the sum of the two
83   /* tails is alpha. For each alpha we can find the
84   /* associated quantile using the SAS function 'quantile'.
85   /* Since alpha is specified, knowing the lower area easily leads to
86   /* the upper area. Here we consider only the lower
87   /* area. As such we can relate interval width as a function of
88   /* lower alpha size. As we increase or decrease the size of the
89   /* lower area we increase or decrease the width of the interval.
90   /* Assuming a minimum interval width exists, we optimize over the
91   /* values [0.00, 0.05].
92   /*
93   /* ***** */
94

```

## Appendix B: Two-Sample SAS Code for Power Simulations

```

95 start alphaLU(ndf1,ddf2,ncp,s1Alpha,tol);
96
97     alphaA = 0;
98     alphaB = s1Alpha;
99
100
101
102
103
104
105
106
107     phi = (-1+sqrt(5))/2;
108     a1 = alphaA+phi*(alphaB-alphaA);
109     wa1 = w(a1,ndf1,ddf2,ncp,s1Alpha);
110     a2 = alphaA+phi*phi*(alphaB-alphaA);
111     wa2 = w(a2,ndf1,ddf2,ncp,s1Alpha);
112
113
114
115
116
117
118
119
120
121     do while (abs(alphaB-alphaA)>tol);
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710

```

## Appendix B: Two-Sample SAS Code for Power Simulations

```

142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188

/* Note 5 ***** */
/*
/* Within the interval [alphaA, a1] we have */
/* the width wa2. a2 is situated by the */
/* expression alphaA+phi*(a1-alphaA). */
/*
/*
/* ***** */

/* Updated alphaB */
/* Save a2 */
/* Save wa2 */
/* Set new test bracket a2 */
/* Calculate interval width for new a2 */

/* If wa1 < wa2 then minimum width is between */
/* a2 and alphaB

/* Note 6 ***** */
/*
/* Within the interval [a2, alphaB] we have */
/* the width wa1. a1 is situated by the */
/* expression alphaA+phi*phi*(alphaB-a2). */
/*
/*
/* ***** */

/* Updated alphaA */
/* Save a1 */
/* Save wa1 */
/* Set new test bracket a1 */
/* Calculate interval width for new a1 */

/* Stopping criteria for GSS method */
/* Number of rows */
/* Number of columns */

/* Sample means */

alphaB = a1;
a1 = a2;
wa1 = wa2;
a2 = alphaA+phi*phi*(alphaB-alphaA);
wa2 = w(a2,ndf1,ddf2,ncp,s1Alpha);
end;
else do;

alphaA = a2;
a2 = a1;
wa2 = wa1;
a1 = alphaA+phi*(alphaB-alphaA);
wa1 = w(a1,ndf1,ddf2,ncp,s1Alpha);
end;
end;

out = (alphaA+alphaB)/2;
return(out);

finish alphaLU;

/* Calculations */

eps = 1e-4;
n = nrow(X);
p = ncol(X);
J = J(p,1);

xBar = X(|+,|)`/n;

```

## Appendix B: Two-Sample SAS Code for Power Simulations

```

189 muG = (&muA+&muB)/p;
190 sst = n*xBar`*xBar-(n/p)*(xBar`*J)**2;
191 df1 = p-1;
192 mst = sst/df1;
193 sse = ssq(X)-(xBar`*xBar)*n;
194 df2 = p*(n-1);
195 mse = sse/df2;
196 fSamp = mst/mse;
197
198 fCrit = quantile('F',1-&alpha,df1,df2);
199
200 pValue = 1-cdf('F',fSamp,df1,df2);
201
202 trueLambda = (n*((&muA-muG)**2)+((&muB-muG)**2))/((&sigma)**2); /* trueLambda */
203
204 truePower = 1-cdf('F',fCrit,df1,df2,trueLambda); /* truePower */
205
206 estLambda = fSamp*(p-1); /* Estimated lambda */
207 adjEstLambda = (estLambda*(df2-2)/df2)-df1; /* Adjusted estimated lambda */
208
209 if adjEstLambda < 0 then adjEstLambda = 0;
210
211 /* Note 7 ***** */
212 /* For some random seeds the */
213 /* noncentrality parameter is */
214 /* negative. */
215 /* A negative noncentrality */
216 /* parameter occurs when */
217 /* the expression */
218 /* estLambda*(df2-2)/df2 is */
219 /* less than df1. */
220 /* ***** */
221
222
223
224 alphaL = alphaU(df1,df2,adjEstLambda,&alpha,eps); /* Lower significance */
225 alphaU = &alpha-alphaL; /* Upper significance */
226
227
228
229
230
231
232
233
234
235

```



236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282

236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282

## Appendix B: Two-Sample SAS Code for Power Simulations

```
283 proc datasets lib=work nolist;  
284     delete sample1 sample2 sampleTwoPops;  
285 quit;  
286 run;  
287  
288 %MEND simulation2;  
289
```

---

## Appendix B: Two-Sample SAS Code for Power Simulations

```

1 /* sampleNormPop2 ***** */
2 /* */
3 /* Program Description: */
4 /* sampleNormPop2 generates a random sample from a normal population */
5 /* and outputs the values to a dataset */
6 /* */
7 /* Input: */
8 /* seed := dataset of random seeds from randSeed */
9 /* popMean := population mean */
10 /* popSD := population standard deviation */
11 /* seedIt := seed value position for generating random observations */
12 /* numObs := sample size */
13 /* sampleNum := identifies sample */
14 /* */
15 /****** */
16
17
18 %macro sampleNormPop2(popMean=, popSD=, seedIt=, numObs=, sampleNum=);
19
20 proc iml;
21   reset nolog;
22   use randSeed;
23   read all into seed;
24
25   x&sampleNum = &popMean+&popSD*rannor(J(&numObs,1,seed[&seedIt,&sampleNum]));
26   test = seed[&seedIt,&sampleNum];
27
28   create sample&sampleNum from x&sampleNum[colname={obs&sampleNum}];
29   append from x&sampleNum;
30
31 quit;
32
33 %mend sampleNormPop2;

```

## Appendix B: Two-Sample SAS Code for Power Simulations

```

1  /* avgMedMultipliesim2 ***** */
2  /*
3  /* Requirement(s):
4  /* avgMedSim2.sas
5  /*
6  /* Program Description:
7  /* avgMedMultipliesim2 calls avgMedSim2 for each sigma by sample size */
8  /* combination.
9  /*
10 /* Input:
11 /* itm          := number of iterations
12 /* nsXnss       := number of iteration groups
13 /*              (sigmas times number of sample sizes)
14 /*
15 /* ***** */
16
17
18 %macro avgMedMultipliesim2(itm=,nsXnss=);
19
20 /* Produce average and median of simulations */
21
22 %let bStart = 1;
23 %let bFinish = &itm;
24
25 %do i = 1 %to &nsXnss %by 1;
26     %avgMedSim2(start=&bStart,finish=&bFinish,itB=&i);
27     %let bStart = &bStart+&itm;
28     %let bFinish = &bFinish+&itm;
29 %end;
30
31
32 /* Combine average simulations into a data set and name columns */
33
34 data avgMedMultipliesimOut;
35 set
36     %do j = 1 %to &nsXnss %by 1;
37         amSim&j
38     %end;
39     ; /* Double do-loop to reference data sets. ';' closes 'set' */
40     rename coll1=mu1 coll2=mu2 coll3=sigma coll4=sampleSize coll5=siglevel coll6=pValue coll7=trueLambda coll8=truePower
41     coll9=estLambda coll10=adjEstLambda coll11=lambdaL coll12=lambdaU coll13=estPower coll14=powerL coll15=powerU
42     coll16=medpower;
43
44     run;
45 %mend avgMedMultipliesim2;

```

## Appendix B: Two-Sample SAS Code for Power Simulations

```

1 /* avgMedSim2 *****/
2 /*
3 /* Program Description:
4 /* avgMedSim2 averages over a user specified number of rows from
5 /* multiplesSimOut. avgMedSim2 also determines the median
6 /* of the calculated power values of the rows.
7 /*
8 /* Input:
9 /* X := gets dataset multiplesSimOut
10 /* start := beginning of iteration block
11 /* finish := end of iteration block
12 /* itB := number of iterations
13 /*
14 /* *****/
15
16
17 %macro avgMedSim2(start=,finish=,itB=);
18
19 proc iml;
20
21     reset nolog;
22     use multiplesSimOut;
23     read all into X;
24
25     /* Average rows for an iteration block */
26
27     rows = X[&start:&finish,];
28     avgRows = rows[,];
29
30
31     /* Median for column calculated power */
32
33     col = X[&start:&finish,13];
34
35     numRows = nrow(col);
36
37     mis = 0;
38     i = 1;
39
40     do while (i <= numRows);
41         if missing((col[i,1])) > 0 then mis=mis//i;
42         i=i+1;
43     end;
44
45
46
47
48     /* Note 1 *****/
49     /* Loop through column of
50     /* of calculated power values.
51     /* If a missing value is found
52     /* record array position.
53     /*
54     /* *****/
55
56     /* Column of calculated power */
57     /* Number of rows */
58     /* Storage for positions of missing values */
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

## Appendix B: Two-Sample SAS Code for Power Simulations

```

48      nr = nrow(mis);
49
50
51      /* Number of missing values */
52
53      /* Note 2 ***** */
54      /* If 'nr' is greater than */
55      /* one then we have missing */
56      /* values. These values are */
57      /* removed from the column. */
58      /* ***** */
59
60      if nr>1 then mis = remove(mis,1);
61      if nr>1 then col = col`;
62      if nr>1 then col = remove(col,mis);
63      if nr>1 then col = col`;
64
65      med = median(col);
66
67
68      /* Concatinate median to average row */
69
70      avgRows=avgRows||med;
71
72
73      create amsim&itB from avgRows;
74      append from avgRows;
75
76      quit;
77      %mend avgMedSim2;

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

1 /* threeSample ***** */
2 /*
3 /* Requirement(s):
4 /* rand3.csv, multiplesim3.sas, avgMedMultiplesim3.sas
5 /*
6 /* Program Description:
7 /* threeSample.sas acts as a controller interface which accepts user
8 /* defined values for three population means, upper and lower bounds
9 /* for a range of population standard deviations, a maximum sample size,
10 /* a type I error rate, and the number of iterations per sample size
11 /* to population standard deviation combination.
12 /*
13 /* ***** */
14
15 proc printto print='\Path\to\Output\Folder\threesampleOut.lst' log='\Path\to\Output\Folder\threesampleOut.log';
16 run;
17
18
19
20
21 /* Input random seed file to the dataset randseed ***** */
22 /*
23 /* rand3.csv is a three column file of random numbers generated from
24 /* http://www.random.org. For each iteration of multiplesim3 three random
25 /* values are used to seed three random samples.
26 /*
27 /* ***** */
28
29 data randSeed;
30   infile "\Path\to\File\rand3.csv" dlm=' ';
31   input seedCol1 seedCol2 seedCol3;
32 run;
33
34
35 /* Produce simulations ***** */
36 /*
37 /* The multiplesim3 manages the production of simulations.
38 /* Calls are made to the macro simulation3.sas which performs
39 /* calculations based on Thomas 1997.
40 /*
41 /* Calculations for each simulation are stored in a dataset called
42 /* multiplesimOut.
43 /*
44 /* ***** */
45
46 %multiplesim3(mu1=30,mu2=40,mu3=35,sigmaMin=5,sigmaMax=40,itMax=5,sampleSizeMax=50,sigLevel=0.05);
47

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

48 /* Print multiplesSimOut */
49
50
51 proc print data=multiplesSimOut;
52 run;
53
54
55 /* Produce averages and medians for simulations ***** */
56 /*
57 /* The avgMedMultiplesSim3 manages the consolidation of simulations.
58 /* Calls are made to avgMedSim3 for each sigma by sample size
59 /* combination. avgMedSim3 averages over a user specified number
60 /* of rows from multiplesSimOut. avgMedSim3 also determines the
61 /* median of the calculated power values of the rows.
62 /*
63 /* Calculations for each iteration block are stored in a dataset
64 /* called avgMedMultiplesSimOut.
65 /*
66 /******
67 %avgMedMultiplesSim3(itM=5,nsXnss=306);
68
69
70
71 /* Print avgMedMultiplesSimOut */
72
73 proc print data=avgMedMultiplesSimOut;
74 run;
75
76
77 /* Plot avgMedMultiplesSimOut */
78
79 goptions ftitle=swiss ftext=swiss;
80 symbol1 value=square interpol=join width=0.5 color=blue;
81 symbol2 value=star interpol=join width=0.5 color=red;
82 symbol3 value=triangle interpol=join width=0.5 color=yellow;
83 symbol4 value='x' interpol=join width=0.5 color=orange;
84 symbol5 value=circle interpol=join width=0.5 color=blue;
85 symbol6 value=dot interpol=join width=0.5 color=green;
86 title2 height=1.4 "Sample Size vs Power";
87 axis1 order=(0.0 to 1.0 by 0.1) label=(angle=90 'Power') minor=none;
88 axis2 order=(5 to 50 by 5) label=(angle=0 'Sample Size');
89 legend label=none value=(h=.8 'Upper AVG Estimated Power' 'True Power' 'AVG Estimated Power' 'Median Power' 'Lower
AVG Estimated Power' 'AVG P-value')
position=center;
90
91
92
93

```



## Appendix B: Three-Sample SAS Code for Power Simulations

```
94 proc gplot data=avgMedMultipleSimOut;  
95   plot powerU*sampleSize truePower*sampleSize estPower*sampleSize medPower*sampleSize powerL*sampleSize  
   pValue*sampleSize/overlay vaxis=axis1 haxis=axis2 hminor=4 legend=legend1;  
96   by sigma;  
97 run;  
98
```

---

## Appendix B: Three-Sample SAS Code for Power Simulations

```

1 /* multiplesim3 ***** */
2 /* */
3 /* Requirement(s): */
4 /* simulation3.sas */
5 /* */
6 /* Program Description: */
7 /* multiplesim3 Iterates simulation3.sas and combines simulation output to a */
8 /* dataset. */
9 /* */
10 /* Input: */
11 /* mu1          := population one mean */
12 /* mu2          := population two mean */
13 /* mu3          := population three mean */
14 /* sigmaMin     := minimum population standard deviation controller */
15 /* sigmaMax     := maximum population standard deviation controller */
16 /* itMax        := number of iterations */
17 /* sampleSizeMax := largest sample size considered */
18 /* sigLevel     := level of significance */
19 /* */
20 /* Note(s): */
21 /* In the subsubloop iterations run in blocks. This is done to */
22 /* traverse the dataset randSeed, where by each call to the macro */
23 /* simulation2 gets a different random seed position. */
24 /* */
25 /* This macro contains two main loops. The first main loop */
26 /* iterates from the minimum user define population standard deviation to 19 */
27 /* by 1. The second main loop iterates from 20 to the maximum user defined */
28 /* population standard deviation by 20. Each main loop has two subloops. */
29 /* The first subloop iterates from a system defined minimum sample size of 5 */
30 /* to 14 by 1. The second subloop iterates from 15 to the maximum user */
31 /* defined sample size by 5. Each subloop has a subsubloop. */
32 /* Each subsubloop iterates by a use defined number of iterations. After */
33 /* a subsubloop executes, the starting position of an iteration 'block' is */
34 /* increased by the the user defined number of iterations. */
35 /* */
36 /* ***** */
37
38 %macro multiplesim3(mu1=,mu2=,mu3=,sigmaMin=,sigmaMax=,itMax=,sampleSizeMax=,sigLevel=);
39
40
41 %local std sd nn it block s n i b;
42 %let block = 1;
43 %let b = 1;
44 %let count = 1;
45
46
47

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

48  /* Produce simulations */
49
50
51  %do std = &sigmaMin %to 19 %by 1;
52    %let sd = %sysevalf(&std/2);
53    %do nn = 5 %to 14 %by 1;
54      %do it = &block %to &block+&itMax-1 %by 1;
55
56      %simulation3(muA=&mul,muB=&mu2,muC=&mu3,sigma=&sd,seedNum=&it,sampleSize=&nn,alpha=&sigLevel,simNum=&count);
57      %let count = %eval(&count+1);
58      %end;
59      %let block = &it;
60      %do nn = 15 %to &sampleSizeMax %by 5;
61        %do it = &block %to &block+&itMax-1 %by 1;
62
63        %simulation3(muA=&mul,muB=&mu2,muC=&mu3,sigma=&sd,seedNum=&it,sampleSize=&nn,alpha=&sigLevel,simNum=&count);
64        %let count = %eval(&count+1);
65        %end;
66        %let block = &it;
67        %end;
68        %do std = 20 %to &sigmaMax %by 20;
69          %let sd = %sysevalf(&std/2);
70          %do nn = 5 %to 14 %by 1;
71            %do it = &block %to &block+&itMax-1 %by 1;
72
73            %simulation3(muA=&mul,muB=&mu2,muC=&mu3,sigma=&sd,seedNum=&it,sampleSize=&nn,alpha=&sigLevel,simNum=&count);
74            %let count = %eval(&count+1);
75            %end;
76            %let block = &it;
77            %do nn = 15 %to &sampleSizeMax %by 5;
78              %do it = &block %to &block+&itMax-1 %by 1;
79
80              %simulation3(muA=&mul,muB=&mu2,muC=&mu3,sigma=&sd,seedNum=&it,sampleSize=&nn,alpha=&sigLevel,simNum=&count);
81              %let count = %eval(&count+1);
82              %end;
83              %let block = &it;
84              %end;
85
86
87
88
89
90

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

91 /* Combine simulations into a dataset and name columns */
92
93 data multipleSimOut;
94   set
95     %do k = 1 %to &count-1 %by 1;
96       sim&k
97     %end;
98   ; /* Double do-loop to reference datasets. ';' closes 'set' */
99
100   rename coll1=mu1 coll2=mu2 coll3=mu3 coll4=sigma coll5=samplesize coll6=sigLevel coll7=pValue coll8=trueLambda
101   col9=truePower coll10=estLambda coll11=adjEstLambda coll12=estLambda coll13=lambdaU coll14=estPower coll15=powerL
102   coll16=powerU;
103
104   run;
105
106 %mend multipleSim3;

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

1 /* simulation3 ***** */
2 /* Requirement(s): */
3 /* sampleNormPop3.sas */
4 /* */
5 /* Program Description: */
6 /* simulation3.sas performs a hypothesis */
7 /* test of no difference for three population means. Random samples */
8 /* are normally distributed with respective means muA, muB and muC with equal */
9 /* population standard deviation sigma. Random samples are generated */
10 /* using seed values from the dataset randSeed. Power calculations */
11 /* are based on methods outlined in Thomas 1997. Calculations are */
12 /* saved to a dataset named 'sim#' where # represents the */
13 /* simulation count. */
14 /* */
15 /* Input: */
16 /* muA := mean of population one */
17 /* muB := mean of population two */
18 /* muC := mean of population three */
19 /* sigma := standard deviation of populations one, two, and three */
20 /* seedNum := position for random seed */
21 /* sampleSize := sample size taken for populations one, two, and three */
22 /* alpha := level of significance */
23 /* simNum := simulation ID */
24 /* */
25 /* Note(s): */
26 /* Optimal lower significance level by Golden Section Search method */
27 /* */
28 /* ***** */
29 /* MACRO simulation3(muA=,muB=,muC=,sigma=,seedNum=,sampleSize=,alpha=,simNum=); */
30 /* */
31 /* Generate random samples from three normal populations */
32 /* */
33 /* sampleNormPop3(popMean=muA,popSD=sigma,seedIt=seedNum,numObs=sampleSize,sampleNum=1); */
34 /* sampleNormPop3(popMean=muB,popSD=sigma,seedIt=seedNum,numObs=sampleSize,sampleNum=2); */
35 /* sampleNormPop3(popMean=muC,popSD=sigma,seedIt=seedNum,numObs=sampleSize,sampleNum=3); */
36 /* */
37 /* Merge datasets sample1, sample2 and sample3 to produce dataset sampleThreePops */
38 /* */
39 data sampleThreePops;
40 merge sample1 sample2 sample3;
41 run;
42
43
44
45
46
47
48

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

49 /* Main */
50
51 proc iml;
52     reset nolog;
53     use sampleThreePops;
54     read all into X;
55
56
57 /* Function ***** */
58 /*
59 /* Return the confidence interval width for a noncentral
60 /* f-dsitribution
61 /*
62 /****** */
63
64     start w(alphaLow,ndf,ddf,ncent,s1);
65
66         width = quantile('F',1-s1+alphaLow,ndf,ddf,ncent)-quantile('F',alphaLow,ndf,ddf,ncent);
67         return(width);
68
69     finish w;
70
71
72 /* Function: optimal lower critical region ***** */
73 /*
74 /* Finds the minimum confidence interval width using an
75 /* implementation of the Golden Section Search method
76 /* similar to Press et al. (2007).
77 /*
78 /* Note(s):
79 /* The Golden Section Search method finds a value which minimizes
80 /* a function. The function must be one dimensional and unimodal.
81 /*
82 /* Imagine a 'typical' pdf of an F distribution. Say we want to
83 /* setup an interval such that the area covered by the sum of the two
84 /* tails is alpha. For each alpha we can find the
85 /* associated quantile using the SAS function 'quantile'.
86 /* Since alpha is specified, knowing the lower area easily leads to
87 /* the upper area. Here we consider only the lower
88 /* area. As such we can relate interval width as a function of
89 /* lower alpha size. As we increase or decrease the size of the
90 /* lower area we increase or decrease the width of the interval.
91 /* Assuming a minimum interval width exists, we optimize over the
92 /* values [0.00, 0.05].
93 /*
94 /****** */
95
96

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

97
98
99
100 start alphaU(ndf1,ddf2,ncp,s1Alpha,tol);
101
102     alphaA = 0;
103     alphaB = s1Alpha;
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144

/* Initial lower bracket on optimal region */
/* Initial upper bracket on optimal region */

/* Note 1 */
/*
/* For each iteration an interval
/* [alphaA, alphaB] is retained
/*
/*
/*
/* Golden ratio constant reduction factor */
/* Set test bracket a1 */
/* Calculate Interval width for a1 */
/* Set test bracket a2 */
/* Calculate interval width for a2 */

/* Note 2 */
/*
/* The intial setup invokes two function
/* calls. The following loop uses one
/* function call per iteration.
/*
/*
/* Note 3 */
/*
/* We continue until the absolute difference
/* between an upper and a lower bracket is
/* negligible. Tolerance is set tole=4.
/*
/*
/* Note 4 */
/*
/* Four brackets are maintained at any
/* given time.
/*
/*
/* If wa1 > wa2 then minimum width is between
/* alphaA and a1

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

145 /* Note 5 ***** */
146 /* Within the interval [alphaA, al] we have */
147 /* the width wa2. a2 is situated by the */
148 /* expression alphaA+phi*(al-alphaA). */
149 /* ***** */
150 /* ***** */
151 /* ***** */
152
153
154 alphaB = al;
155 a1 = a2;
156 wa1 = wa2;
157 a2 = alphaA+phi*phi*(alphaB-alphaA);
158 wa2 = w(a2,ndf1,ddf2,ncp,s1Alpha);
159 end;
160 else do;
161
162
163
164
165
166
167
168
169
170
171 alphaA = a2;
172 a2 = a1;
173 wa2 = wa1;
174 a1 = alphaA+phi*phi*(alphaB-alphaA);
175 wa1 = w(a1,ndf1,ddf2,ncp,s1Alpha);
176 end;
177
178
179 out = (alphaA+alphaB)/2;
180 return(out);
181
182 finish alphaU;
183
184
185 /* Calculations */
186
187 eps = le-4;
188 n = nrow(X);
189 p = ncol(X);
190 J = J(p,1);
191 muG = (&muA+&muB+&muC)/p;
192 xBar = X(|+,|)^`/n;

```

```

/* Note 5 ***** */
/* Within the interval [alphaA, al] we have */
/* the width wa2. a2 is situated by the */
/* expression alphaA+phi*(al-alphaA). */
/* ***** */
/* ***** */
/* ***** */

/* Updated alphaB */
/* Save a2 */
/* Save wa2 */
/* Set new test bracket a2 */
/* Calculate interval width for new a2 */

/* If wa1 < wa2 then minimum width */
/* is between a2 and alphaB

/* Note 6 ***** */
/* Within the interval [a2, alphaB] we have */
/* the width wa1. al is situated by the */
/* expression alphaA+phi*phi*(alphaB-a2). */
/* ***** */
/* ***** */
/* ***** */

/* Updated alphaA */
/* Save al */
/* Save wa1 */
/* Set new test bracket al */
/* Calculate interval width for new al */

/* Stopping criteria for GSS method */
/* Number of rows */
/* Number of columns */

/* Mean of population means */
/* Sample means */

```



## Appendix B: Three-Sample SAS Code for Power Simulations

```

193 sst = n*xBar`*xBar-(n/p)*(xBar`*J)**2;
194 df1 = p-1;
195 mst = sst/df1;
196 sse = ssq(X)-(xBar`*xBar)*n;
197 df2 = p*(n-1);
198 mse = sse/df2;
199 fSamp = mst/mse;
200
201 fCrit = quantile('F',1-&alpha,df1,df2);
202
203 pValue = 1-cdf('F',fSamp,df1,df2);
204
205 trueLambda = (n*(((&muA-muG)**2)+((&muB-muG)**2)+((&muC-muG)**2)))/((&sigma)**2); /* trueLambda */
206
207 truePower = 1-cdf('F',fCrit,df1,df2,trueLambda);
208 /* truePower */
209
210 estLambda = fSamp*(p-1);
211 adjEstLambda = (estLambda*(df2-2)/(df2)-df1);
212
213 if adjEstLambda < 0 then adjEstLambda = 0;
214
215 /* Note 7 ***** */
216 /* For some random seeds the */
217 /* noncentrality parameter is */
218 /* negative. */
219 /* A negative noncentrality */
220 /* parameter occurs when */
221 /* the expression */
222 /* estLambda*(df2-2)/df2 is */
223 /* less than df1. */
224 /* ***** */
225
226
227 alphaL = alphaU(df1,df2,adjEstLambda,&alpha,eps);
228 alphaU = &alpha-alphaL;
229
230
231
232
233
234
235
236
237
238
239
240

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

241 adjAlphaL = 1-alphaL;
242 probCenF = cdf('F',fsamp,df1,df2);
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288

/* Note 8 *****/
/*
/* fnonct reports an error for a supplied */
/* probability greater than the */
/* probability from the cdf of a central */
/* F distribution.
*/
/* As the probability supplied gets large */
/* fnonct tends toward zero. probCenF */
/* with fnonct is small around zero. */
/* Using adjAlphaL would then be smaller */
/* about zero. Set to zero.
*/
/* *****/

/* See note 8 */
/* See note 8 */

/* Estimated power */
/* Lower limit for power */
/* Upper limit for power */

/* Output simulation to a data set */

Y = {'&muA' '&muB' '&muC' '&sigma' '&sampleSize' '&alpha' ". ". ". ". ". ". ". ". ". ". ". "};
Y = num(Y);
Y[7] = pValue;
Y[8] = trueLambda;
Y[9] = truePower;
Y[10] = estLambda;
Y[11] = adjEstLambda;
Y[12] = lambdaL;
Y[13] = lambdaU;
Y[14] = estPower;
Y[15] = powerL;
Y[16] = powerU;

create sim&simNum from Y;
append from Y;

quit;

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```
289 proc datasets lib=work nolist;  
290   delete sample1 sample2 sample3 sampleThreePops;  
291   quit;  
292   run;  
293  
294   %MEND simulation3;  
295
```

---

## Appendix B: Three-Sample SAS Code for Power Simulations

```

1 /* sampleNormPop3 ***** */
2 /* */
3 /* Program Description: */
4 /* sampleNormPop3 generates a random sample from a normal population */
5 /* and outputs the values to a dataset */
6 /* */
7 /* Input: */
8 /* seed := dataset of random seeds from randSeed */
9 /* popMean := population mean */
10 /* popSD := population standard deviation */
11 /* seedIt := seed value position for generating random observations */
12 /* numObs := sample size */
13 /* sampleNum := identifies sample */
14 /* */
15 /****** */
16
17
18 %macro sampleNormPop3(popMean=, popSD=, seedIt=, numObs=, sampleNum=);
19
20 proc iml;
21   reset nolog;
22   use randSeed;
23   read all into seed;
24
25   x&sampleNum = &popMean+&popSD*rannor(J(&numObs,1,seed[&seedIt,&sampleNum]));
26   test = seed[&seedIt,&sampleNum];
27
28   create sample&sampleNum from x&sampleNum[colname={obs&sampleNum}];
29   append from x&sampleNum;
30
31 quit;
32
33 %mend sampleNormPop3;

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

1 /* avgMedMultipliesim3 ***** */
2 /*
3 /* Requirement(s):
4 /* avgMedSim3.sas
5 /*
6 /* Program Description:
7 /* avgMedMultipliesim3 calls avgMedSim3 for each sigma by sample size */
8 /* combination.
9 /*
10 /* Input:
11 /* itm          := number of iterations
12 /* nsXnss       := number of iteration groups
13 /*              (sigmas times number of sample sizes)
14 /*
15 /* ***** */
16
17
18 %macro avgMedMultipliesim3(itm=,nsXnss=);
19
20 /* Produce average and median of simulations */
21
22 %let bStart = 1;
23 %let bFinish = &itm;
24
25 %do i = 1 %to &nsXnss %by 1;
26     %avgMedSim3(start=&bStart,finish=&bFinish,itB=&i);
27     %let bStart = &bStart+&itm;
28     %let bFinish = &bFinish+&itm;
29 %end;
30
31
32 /* Combine average simulations into a data set and name columns */
33
34 data avgMedMultipliesimOut;
35 set
36     %do j = 1 %to &nsXnss %by 1;
37         amSim&j
38     %end;
39 ; /* Double do-loop to reference data sets. ';' closes 'set' */
40 rename coll1=mu1 coll2=mu2 coll3=mu3 coll4=sigma coll5=sampleSize coll6=sigLevel coll7=pValue coll8=trueLambda
... coll9=truePower coll10=estLambda coll11=adjEstLambda coll12=lambdaL coll13=lambdaU coll14=estPower coll15=powerL
coll16=powerU coll17=medPower;
41
42 run;
43
44 %mend avgMedMultipliesim3;
45

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

1 /* avgMedSim3 ***** */
2 /*
3 /* Program Description:
4 /* avgMedSim3 averages over a user specified number of rows from
5 /* multiplesimOut. avgMedSim3 also determines the median
6 /* of the calculated power values of the rows.
7 /*
8 /* Input:
9 /* X := gets dataset multiplesimOut
10 /* start := beginning of iteration block
11 /* finish := end of iteration block
12 /* itB := number of iterations
13 /*
14 /* ***** */
15
16
17 %macro avgMedSim3(start=,finish=,itB=);
18
19 proc iml;
20
21 reset nolog;
22 use multiplesimOut;
23 read all into X;
24
25 /* Average rows for an iteration block */
26
27 rows = X[&start:&finish,];
28 avgRows = rows[:,];
29
30
31 /* Median for column calculated power */
32
33 col = X[&start:&finish,14];
34
35 numRows = nrow(col);
36
37 mis = 0;
38 i = 1;
39
40 do while (i <= numRows);
41 if missing((col[i,1])) > 0 then mis=mis//i;
42 i=i+1;
43 end;
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

## Appendix B: Three-Sample SAS Code for Power Simulations

```

48      nr = nrow(mis);
49
50
51      /* Number of missing values */
52
53      /* Note 2 ***** */
54      /* If 'nr' is greater than */
55      /* one then we have missing */
56      /* values. These values are */
57      /* removed from the column. */
58      /* ***** */
59
60      if nr>1 then mis = remove(mis,1);
61      if nr>1 then col = col`;
62      if nr>1 then col = remove(col,mis);
63      if nr>1 then col = col`;
64
65      med = median(col);
66
67
68      /* Concatinate median to average row */
69
70      avgRows=avgRows||med;
71
72
73      create amsim&itB from avgRows;
74      append from avgRows;
75
76      quit;
77      %mend avgMedSim3;

```

## Appendix B: Four-Sample SAS Code for Power Simulations

```

1 /* fourSample ***** */
2 /* */
3 /* Requirement(s): */
4 /* rand4.csv, multiplesim4.sas, avgMedMultipleSim4.sas */
5 /* */
6 /* Program Description: */
7 /* fourSample.sas acts as a controller interface which accepts user */
8 /* defined values for four population means, upper and lower bounds */
9 /* for a range of population standard deviations, a maximum sample size, */
10 /* a type I error rate, and the number of iterations per sample size */
11 /* to population standard deviation combination. */
12 /* */
13 /****** */
14
15
16 proc printto printto='\Path\to\Output\Folder\fourSampleOut.lst'
17             log='\Path\to\Output\Folder\fourSampleOut.log';
18 run;
19
20
21
22 /* Input random seed file to the dataset randseed ***** */
23 /* */
24 /* rand4.csv is a four column file of random numbers generated from */
25 /* http://www.random.org. For each iteration of multiplesim4 four random */
26 /* values are used to seed four random samples. */
27 /* ***** */
28
29 data randseed;
30     infile "\Path\to\File\rand4.csv" dlm=' ';
31     input seedCol1 seedCol2 seedCol3 seedCol4;
32 run;
33
34
35
36 /* Produce simulations ***** */
37 /* */
38 /* The multiplesim4 manages the production of simulations. */
39 /* Calls are made to the macro simulation4.sas which performs */
40 /* calculations based on Thomas 1997. */
41 /* */
42 /* Calculations for each simulation are stored in a dataset called */
43 /* multiplesimOut. */
44 /* ***** */
45
46 %multiplesim4(mu1=30,mu2=40,mu3=35,mu4=35,sigmaMin=5,sigmaMax=40,itMax=100,sampleSizeMax=50,sigLevel=0.05);

```



## Appendix B: Four-Sample SAS Code for Power Simulations

```

48
49
50 /* Print multiplesimOut */
51
52 proc print data=multiplesimOut;
53 run;
54
55 /* Produce averages and medians for simulations *****/
56 /*
57 /* The avgMedMultipleSim4 manages the consolidation of simulations. */
58 /* Calls are made to avgMedSim4 for each sigma by sample size */
59 /* combination. avgMedSim4 averages over a user specified number */
60 /* of rows from multiplesimOut. avgMedSim4 also determines the */
61 /* median of the calculated power values of the rows. */
62 /*
63 /*
64 /* Calculations for each iteration block are stored in a dataset */
65 /* called avgMedMultipleSimOut. */
66 /*
67 /******
68 %avgMedMultipleSim4(itM=100,nsXnss=306);
69
70
71
72 /* Print avgMedMultipleSimOut */
73
74 proc print data=avgMedMultipleSimOut;
75 run;
76
77
78 /* Plot avgMedMultipleSimOut */
79
80 options ftitle=swiss ftext=swiss;
81 symbol1 value=square interpol=join width=0.5 color=blue;
82 symbol2 value=star interpol=join width=0.5 color=red;
83 symbol3 value=triangle interpol=join width=0.5 color=yellow;
84 symbol4 value='x' interpol=join width=0.5 color=orange;
85 symbol5 value=circle interpol=join width=0.5 color=blue;
86 symbol6 value=dot interpol=join width=0.5 color=green;
87 title2 height=1.4 "Sample Size vs Power";
88 axis1 order=(0.0 to 1.0 by 0.1) label=(angle=90 'Power') minor=none;
89 axis2 order=(5 to 50 by 5) label=(angle=0 'Sample Size');
90 legend label=none value=(h=.8 'Upper AVG Estimated Power' 'True Power' 'AVG Estimated Power' 'Median Power'
91 'Lower AVG Estimated Power' 'AVG p-Value')
92 position=center;
93
94

```

## Appendix B: Four-Sample SAS Code for Power Simulations

```
95 proc gplot data=avgMedMultipleSimOut;  
96   plot powerU*sampleSize truePower*sampleSize estPower*sampleSize medPower*sampleSize powerL*sampleSize  
97   pValue*sampleSize/overlay vaxis=axis1 haxis=axis2 hminor=4 legend=legend1;  
98   by sigma;  
99 run;  
100
```

---

## Appendix B: Four-Sample SAS Code for Power Simulations

```

1 /* multiplesim4 ***** */
2 /*
3 /* Requirement(s):
4 /* simulation4.sas
5 /*
6 /* Program Description:
7 /* multiplesim4 Iterates simulation4.sas and combines simulation output to a
8 /* dataset.
9 /*
10 /* Input:
11 /* mu1          := population one mean
12 /* mu2          := population two mean
13 /* mu3          := population three mean
14 /* mu4          := population four mean
15 /* singaMin      := minimum population standard deviation controller
16 /* sigmaMax      := maximum population standard deviation controller
17 /* itMax         := number of iterations
18 /* sampleSizeMax := largest sample size considered
19 /* sigLevel      := level of significance
20 /*
21 /* Note(s):
22 /* In the subsubloop iterations run in blocks. This is done to
23 /* traverse the dataset randSeed, where by each call to the macro
24 /* simulation4 gets a different random seed position.
25 /*
26 /* This macro contains two main loops. The first main loop
27 /* iterates from the minimum user define population standard deviation to 19
28 /* by 1. The second main loop iterates from 20 to the maximum user defined
29 /* population standard deviation by 20. Each main loop has two subloops.
30 /* The first subloop iterates from a system defined minimum sample size of 5
31 /* to 14 by 1. The second subloop iterates from 15 to the maximum user
32 /* defined sample size by 5. Each subloop has a subsubloop.
33 /* Each subsubloop iterates by a use defined number of iterations. After
34 /* a subsubloop executes, the starting position of an iteration 'block' is
35 /* increased by the the user defined number of iterations.
36 /*
37 /****** */
38
39 %macro multiplesim4(mu1=,mu2=,mu3=,mu4=,sigmaMin=,sigmaMax=,itMax=,sampleSizeMax=,sigLevel=);
40
41 %local std sd nn it block s n i b;
42 %let block = 1;
43 %let b = 1;
44 %let count = 1;
45
46
47

```

## Appendix B: Four-Sample SAS Code for Power Simulations

```

48 /* Produce simulations */
49
50 %do std = &sigmaMin %to 19 %by 1;
51   %let sd = %sysevalf(&std/2);
52   %do nn = 5 %to 14 %by 1;
53     %do it = &bblock %to &bblock+&itMax-1 %by 1;
54       %simulation4(muA=&m1,muB=&mu2,muC=&mu3,muD=&mu4,sigma=&sds,seedNum=&it,sampleSize=&nns,alpha=&sigLevel,simNum=&count)
55 ;
56   %let count = %eval(&count+1);
57   %end;
58   %let bblock = &it;
59   %end;
60   %do nn = 15 %to &samplesizeMax %by 5;
61     %do it = &bblock %to &bblock+&itMax-1 %by 1;
        %simulation4(muA=&m1,muB=&mu2,muC=&mu3,muD=&mu4,sigma=&sds,seedNum=&it,sampleSize=&nns,alpha=&sigLevel,simNum=&count)
62 ;
63   %let count = %eval(&count+1);
64   %end;
65   %let bblock = &it;
66   %end;
67   %do std = 20 %to &sigmaMax %by 20;
68     %let sd = %sysevalf(&std/2);
69     %do nn = 5 %to 14 %by 1;
70       %do it = &bblock %to &bblock+&itMax-1 %by 1;
71         %simulation4(muA=&m1,muB=&mu2,muC=&mu3,muD=&mu4,sigma=&sds,seedNum=&it,sampleSize=&nns,alpha=&sigLevel,simNum=&count)
72 ;
73     %let count = %eval(&count+1);
74     %end;
75     %let bblock = &it;
76     %end;
77     %do nn = 15 %to &samplesizeMax %by 5;
78       %do it = &bblock %to &bblock+&itMax-1 %by 1;
        %simulation4(muA=&m1,muB=&mu2,muC=&mu3,muD=&mu4,sigma=&sds,seedNum=&it,sampleSize=&nns,alpha=&sigLevel,simNum=&count)
79 ;
80     %let count = %eval(&count+1);
81     %end;
82     %let bblock = &it;
83     %end;
84     %end;
85   %end;
86 
```

## Appendix B: Four-Sample SAS Code for Power Simulations

```
87 /* Combine simulations into a dataset and name columns */
88
89 data multipleSimOut;
90   set
91     %do k = 1 %to &count-1 %by 1;
92       sim&k
93     %end;
94   ; /* Double do-loop to reference datasets. ';' closes 'set' */
95
96   rename coll1=mu1 coll2=mu2 coll3=mu3 coll4=mu4 coll5=sigma coll6=sampleSize coll7=sigLevel coll8=pValue
97   coll9=trueLambda coll10=truePower coll11=estLambda coll12=adjEstLambda coll13=lambdaL coll14=lambdaU coll15=estPower
98   coll16=powerL coll17=powerU;
99
100   run;
101
102 %mend multipleSim4;
```

## Appendix B: Four-Sample SAS Code for Power Simulations

```

1  /* simulation4 ***** */
2  /*
3  /* Requirement(s):
4  /* sampleNormPop4.sas
5  /*
6  /* Program Description:
7  /* simulation4.sas performs a hypothesis
8  /* test of no difference for four population means. Random samples
9  /* are normally distributed with respective means muA, muB, muC and muD
10 /* with equal population standard deviation sigma. Random samples are
11 /* generated using seed values from the dataset randSeed. Power
12 /* calculations are based on methods outlined in Thomas 1997.
13 /* Calculations are saved to a dataset named 'sim#' where # represents
14 /* the simulation count.
15 /*
16 /* Input:
17 /* muA      := mean of population one
18 /* muB      := mean of population two
19 /* muC      := mean of population three
20 /* muD      := mean of population four
21 /* sigma    := standard deviation of populations one and two
22 /* seedNum  := position for random seed
23 /* sampleSize := sample size taken for populations one and two
24 /* alpha    := level of significance
25 /* simNum   := simulation ID
26 /*
27 /* Note(s):
28 /* Optimal lower significance level by Golden Section Search method
29 /*
30 /* ***** */
31
32
33 %MACRO simulation4(muA=,muB=,muC=,muD=,sigma=,seedNum=,sampleSize=,alpha=,simNum=);
34
35     /* Generate random samples from four normal populations */
36
37     %sampleNormPop4(popMean=&muA,popSD=&sigma,seedIt=&seedNum,numObs=&sampleSize,sampleNum=1);
38     %sampleNormPop4(popMean=&muB,popSD=&sigma,seedIt=&seedNum,numObs=&sampleSize,sampleNum=2);
39     %sampleNormPop4(popMean=&muC,popSD=&sigma,seedIt=&seedNum,numObs=&sampleSize,sampleNum=3);
40     %sampleNormPop4(popMean=&muD,popSD=&sigma,seedIt=&seedNum,numObs=&sampleSize,sampleNum=4);
41
42
43     /* Merge datasets sample1, sample2, sample3 and sample4 to produce dataset sampleFourPops */
44
45     data sampleFourPops;
46         merge sample1 sample2 sample3 sample4;
47     run;

```

## Appendix B: Four-Sample SAS Code for Power Simulations

```

48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94

/* Main */
proc iml;
    reset nolog;
    use sampleFourPops;
    read all into X;

    /* Function *****/
    /*
    /* Return the confidence interval width for a noncentral
    /* f-distribution
    /*
    /* *****/
    start w(alphaLow,ndf,ddf,ncent,sl);

        width = quantile('F',1-sl+alphaLow,ndf,ddf,ncent)-quantile('F',alphaLow,ndf,ddf,ncent);
        return(width);

    finish w;

    /* Function: optimal lower critical region *****/
    /*
    /* Finds the minimum confidence interval width using an
    /* implementation of the Golden Section Search method
    /* similar to Press et al. (2007).
    /*
    /* Note(s):
    /* The Golden Section Search method finds a value which minimizes
    /* a function. The function must be one dimensional and unimodal.
    /*
    /* Imagine a 'typical' pdf of an F distribution. Say we want to
    /* setup an interval such that the area covered by the sum of the two
    /* tails is alpha. For each alpha we can find the
    /* associated quantile using the SAS function 'quantile'.
    /* Since alpha is specified, knowing the lower area easily leads to
    /* the upper area. Here we consider only the lower
    /* area. As such we can relate interval width as a function of
    /* lower alpha size. As we increase or decrease the size of the
    /* lower area we increase or decrease the width of the interval.
    /* Assuming a minimum interval width exists, we optimize over the
    /* values [0.00, 0.05].
    /*
    /* *****/

```

95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141

```

if wa1>wa2 then do;
/* If wa1 > wa2 then minimum width is between */
/* alphaA and a1 */

```



## Appendix B: Four-Sample SAS Code for Power Simulations

```

142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188

/* Note 5 ***** */
/*
/* Within the interval [alphaA, a1] we have */
/* the width wa2. a2 is situated by the */
/* expression alphaA+phi*(a1-alphaA). */
/*
/*
/* Updated alphaB */
/* Save a2 */
/* Save wa2 */
/* Set new test bracket a2 */
/* Calculate interval width for new a2 */

/* If wa1 < wa2 then minimum width is between */
/* a2 and alphaB

/* Note 6 ***** */
/*
/* Within the interval [a2, alphaB] we have */
/* the width wa1. a1 is situated by the */
/* expression alphaA+phi*phi*(alphaB-a2). */
/*
/*
/* Updated alphaA */
/* Save a1 */
/* Save wa1 */
/* Set new test bracket a1 */
/* Calculate interval width for new a1 */

/* Stopping criteria for GSS method */
/* Number of rows */
/* Number of columns */

/* Mean of population means */
/* Sample means */

alphaB = a1;
a1 = a2;
wa1 = wa2;
a2 = alphaA+phi*phi*(alphaB-alphaA);
wa2 = w(a2,ndf1,ddf2,ncp,s1Alpha);
end;
else do;

alphaA = a2;
a2 = a1;
wa2 = wa1;
a1 = alphaA+phi*(alphaB-alphaA);
wa1 = w(a1,ndf1,ddf2,ncp,s1Alpha);
end;
end;

out = (alphaA+alphaB)/2;
return(out);

finish alphaLU;

/* Calculations */

eps = 1e-4;
n = nrow(X);
p = ncol(X);
J = J(p,1);
muG = (&muA+&muB+&muC+&muD)/p;
xBar = X(|+,|)`/n;

```

## Appendix B: Four-Sample SAS Code for Power Simulations

```

189 sst = n*xBar`*xBar-(n/p)*(xBar`*J)**2;
190 df1 = p-1;
191 mst = sst/df1;
192 sse = ssq(X)-(xBar`*xBar)*n;
193 df2 = p*(n-1);
194 mse = sse/df2;
195 fSamp = mst/mse;
196
197 fCrit = quantile('F',1-&alpha,df1,df2);
198
199 pValue = 1-cdf('F',fSamp,df1,df2);
200
201 trueLambda = (n*((&muA-muG)**2)+(&muB-muG)**2)+((&muC-muG)**2)+((&muD-muG)**2))/((&sigma)**2);
202
203
204 truePower = 1-cdf('F',fCrit,df1,df2,trueLambda);
205
206 estLambda = fSamp*(p-1);
207 adjEstLambda = (estLambda*(df2-2)/df2)-df1;
208
209 if adjEstLambda < 0 then adjEstLambda = 0;
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235

```

```

/* Sum of squares for treatments */
/* Degrees of freedom for treatment */
/* Mean square for treatment */
/* Sum of squares for error */
/* Degrees of freedom for error */
/* Mean square for error */
/* Observed F */

/* F critical value */

/* P-Value */

/* Note 7 */
/* For some random seeds the
/* noncentrality parameter is
/* negative.
/*
/* A negative noncentrality
/* parameter occurs when
/* the expression
/* estLambda*(df2-2)/df2 is
/* less than df1.
/*
/* Lower significance */
/* Upper significance */

```

## Appendix B: Four-Sample SAS Code for Power Simulations

```

236 adjAlphaL = 1-alphaL;
237 probcenF = cdf('F',fSamp,df1,df2);
238
239 /* Note 8 ***** */
240 /* fnonct reports an error for a supplied */
241 /* probability greater than the */
242 /* probability from the cdf of a central */
243 /* F distribution. */
244 /* As the probability supplied gets large */
245 /* fnonct tends toward zero. probCenF */
246 /* with fnonct is small around zero. */
247 /* Using adjAlphaL would then be smaller */
248 /* about zero. Set to zero. */
249 /* ***** */
250
251
252 /* See note 8 */
253
254 /* See note 8 */
255
256
257
258 if adjAlphaL > probcenF then lambdaL = 0;
259 else lambdaL = fnonct(fSamp,df1,df2,adjAlphaL);
260
261 if alphaU > probcenF then lambdaU = 0;
262 else lambdaU = fnonct(fSamp,df1,df2,alphaU);
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

## Appendix B: Four-Sample SAS Code for Power Simulations

```
283 proc datasets lib=work nolist;  
284     delete sample1 sample2 sample3 sample4 sampleFourPops;  
285 quit;  
286 run;  
287  
288 %MEND simulation4;  
289
```

---

## Appendix B: Four-Sample SAS Code for Power Simulations

```

1 /* sampleNormPop4 ***** */
2 /* */
3 /* Program Description: */
4 /* sampleNormPop4 generates a random sample from a normal population */
5 /* and outputs the values to a dataset */
6 /* */
7 /* Input: */
8 /* seed := dataset of random seeds from randSeed */
9 /* popMean := population mean */
10 /* popSD := population standard deviation */
11 /* seedIt := seed value position for generating random observations */
12 /* numObs := sample size */
13 /* sampleNum := identifies sample */
14 /* */
15 /****** */
16
17
18 %macro sampleNormPop4(popMean=, popSD=, seedIt=, numObs=, sampleNum=);
19
20 proc iml;
21   reset nolog;
22   use randSeed;
23   read all into seed;
24
25   x&sampleNum = &popMean+&popSD*rannor(J(&numObs,1,seed[&seedIt,&sampleNum]));
26   test = seed[&seedIt,&sampleNum];
27
28   create sample&sampleNum from x&sampleNum[colname={obs&sampleNum}];
29   append from x&sampleNum;
30
31 quit;
32
33 %mend sampleNormPop4;

```

## Appendix B: Four-Sample SAS Code for Power Simulations

```

1  /* avgMedMultipliesim4 ***** */
2  /*
3  /* Requirement(s):
4  /* avgMedSim4.sas
5  /*
6  /* Program Description:
7  /* avgMedMultipliesim4 calls avgMedSim4 for each sigma by sample size */
8  /* combination.
9  /*
10 /* Input:
11 /* itm          := number of iterations
12 /* nsXnss       := number of iteration groups
13 /*              (sigmas times number of sample sizes)
14 /*
15 /* ***** */
16
17
18 %macro avgMedMultipliesim4(itm=,nsXnss=);
19
20 /* Produce average and median of simulations */
21
22 %let bStart = 1;
23 %let bFinish = &itm;
24
25 %do i = 1 %to &nsXnss %by 1;
26     %avgMedSim4(start=&bStart,finish=&bFinish,itB=&i);
27     %let bStart = &bStart+&itm;
28     %let bFinish = &bFinish+&itm;
29 %end;
30
31
32 /* Combine average simulations into a data set and name columns */
33
34 data avgMedMultipliesimOut;
35 set
36     %do j = 1 %to &nsXnss %by 1;
37         amSim&j
38     %end;
39 ; /* Double do-loop to reference data sets. ';' closes 'set' */
40 rename coll1=mu1 coll2=mu2 coll3=mu3 coll4=mu4 coll5=sigma coll6=sampleSize coll7=sigLevel coll8=pValue
... coll9=trueLambda coll10=truePower coll11=estLambda coll12=adjEstLambda coll13=lambda coll14=lambdaU coll15=estPower
coll16=powerL coll17=powerU coll18=medPower;
41
42 run;
43
44 %mend avgMedMultipliesim4;
45

```

## Appendix B: Four-Sample SAS Code for Power Simulations

```

1 /* avgMedSim4 *****/
2 /*
3 /* Program Description:
4 /* avgMedSim4 averages over a user specified number of rows from
5 /* multipleSimOut. avgMedSim4 also determines the median
6 /* of the calculated power values of the rows.
7 /*
8 /* Input:
9 /* X := gets dataset multipleSimOut
10 /* start := beginning of iteration block
11 /* finish := end of iteration block
12 /* itB := number of iterations
13 /*
14 /* *****/
15
16
17 %macro avgMedSim4(start=,finish=,itB=);
18
19 proc iml;
20
21 reset nolog;
22 use multipleSimOut;
23 read all into X;
24
25 /* Average rows for an iteration block */
26
27 rows = X[&start:&finish,];
28 avgRows = rows[,];
29
30
31 /* Median for column calculated power */
32
33 col = X[&start:&finish,15];
34
35 numRows = nrow(col);
36
37 mis = 0;
38 i = 1;
39
40 do while (i <= numRows);
41 if missing((col[i,1])) > 0 then mis=mis//i;
42 i=i+1;
43 end;
44
45
46
47

```

/\* Column of calculated power \*/  
 /\* Number of rows \*/  
 /\* Storage for positions of missing values \*/  
 /\* Note 1 \*\*\*\*\*/  
 /\* Loop through column of  
 /\* of calculated power values. \*/  
 /\* If a missing value is found \*/  
 /\* record array position. \*/  
 /\*  
 /\* \*\*\*\*\*/

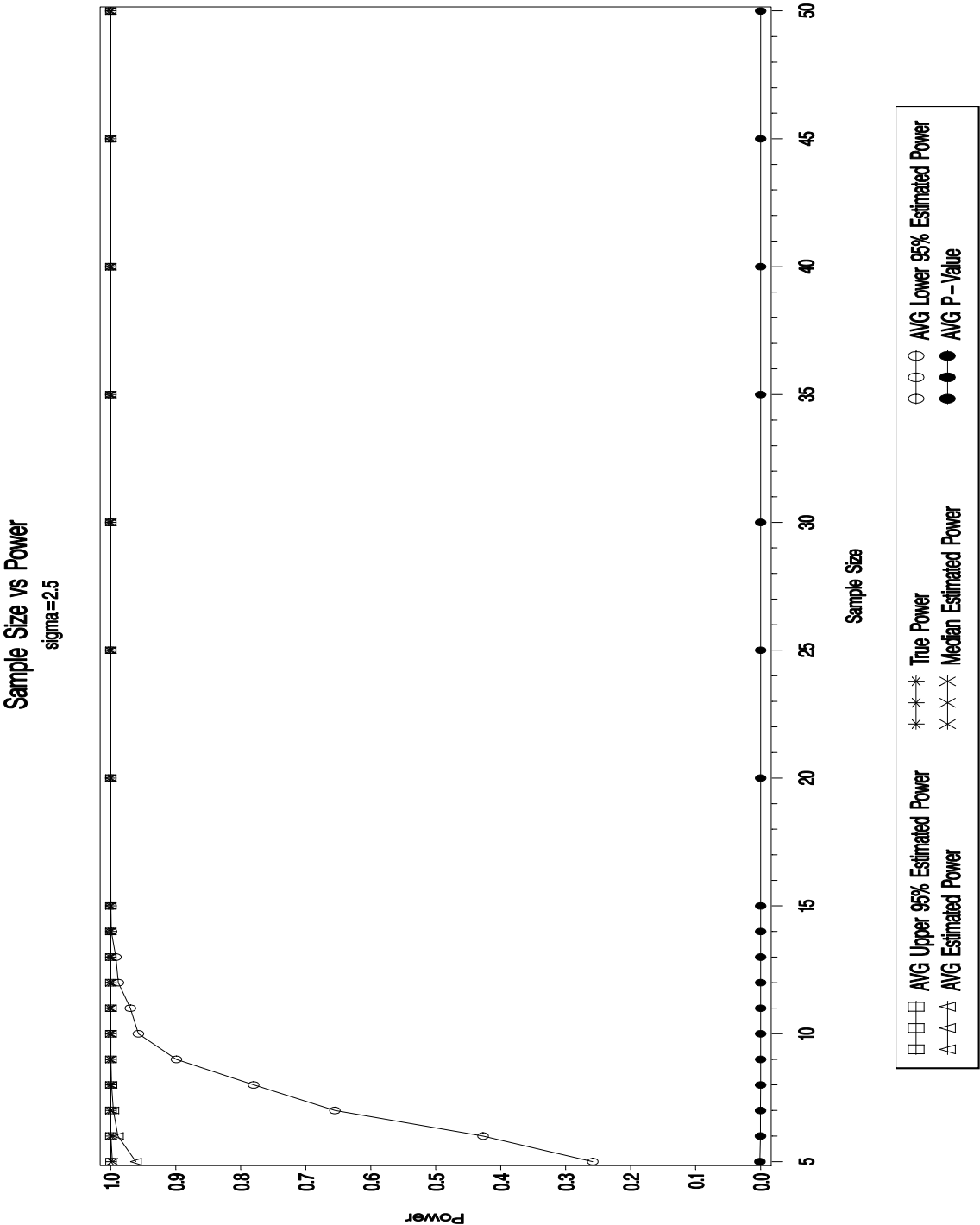
## Appendix B: Four-Sample SAS Code for Power Simulations

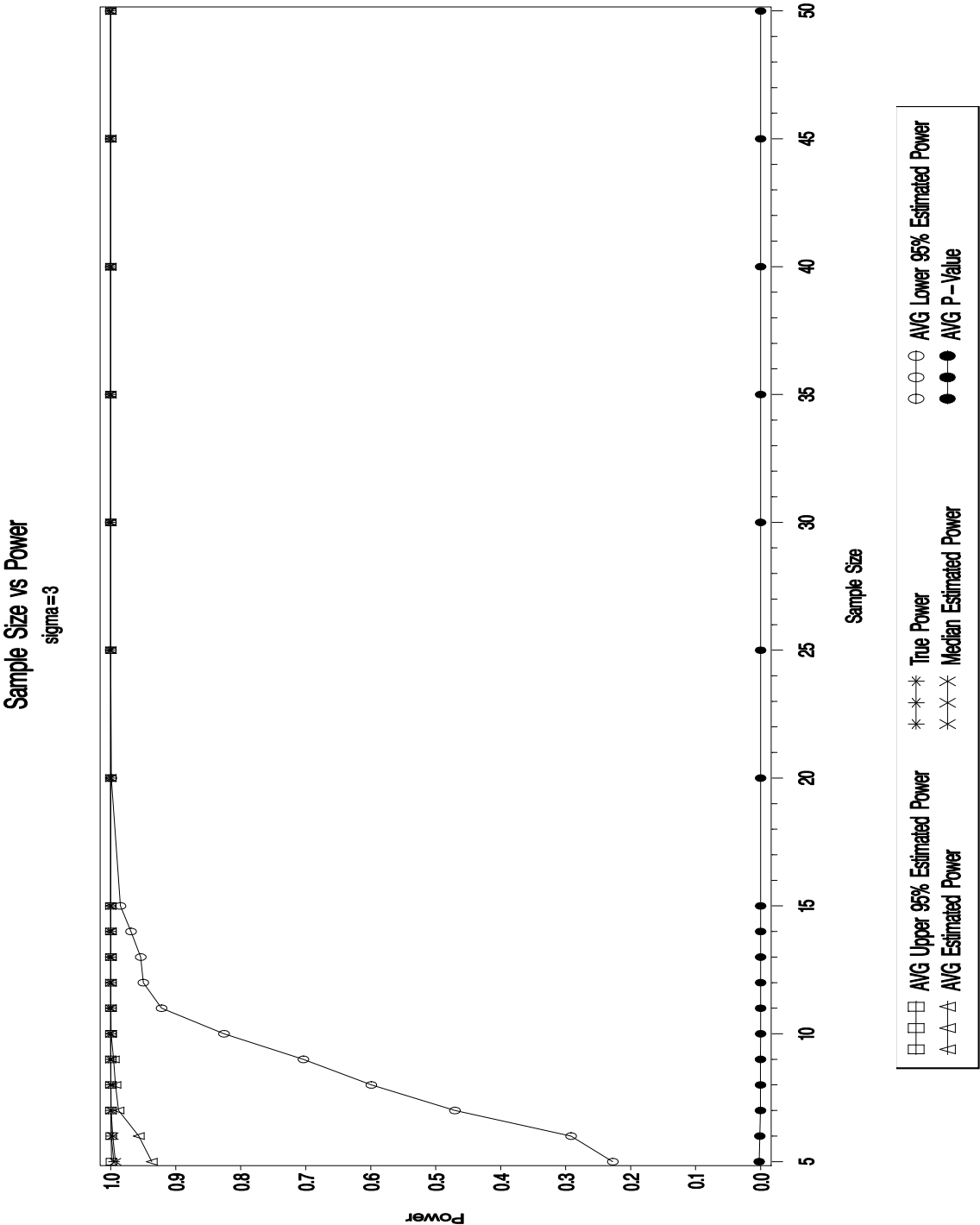
```

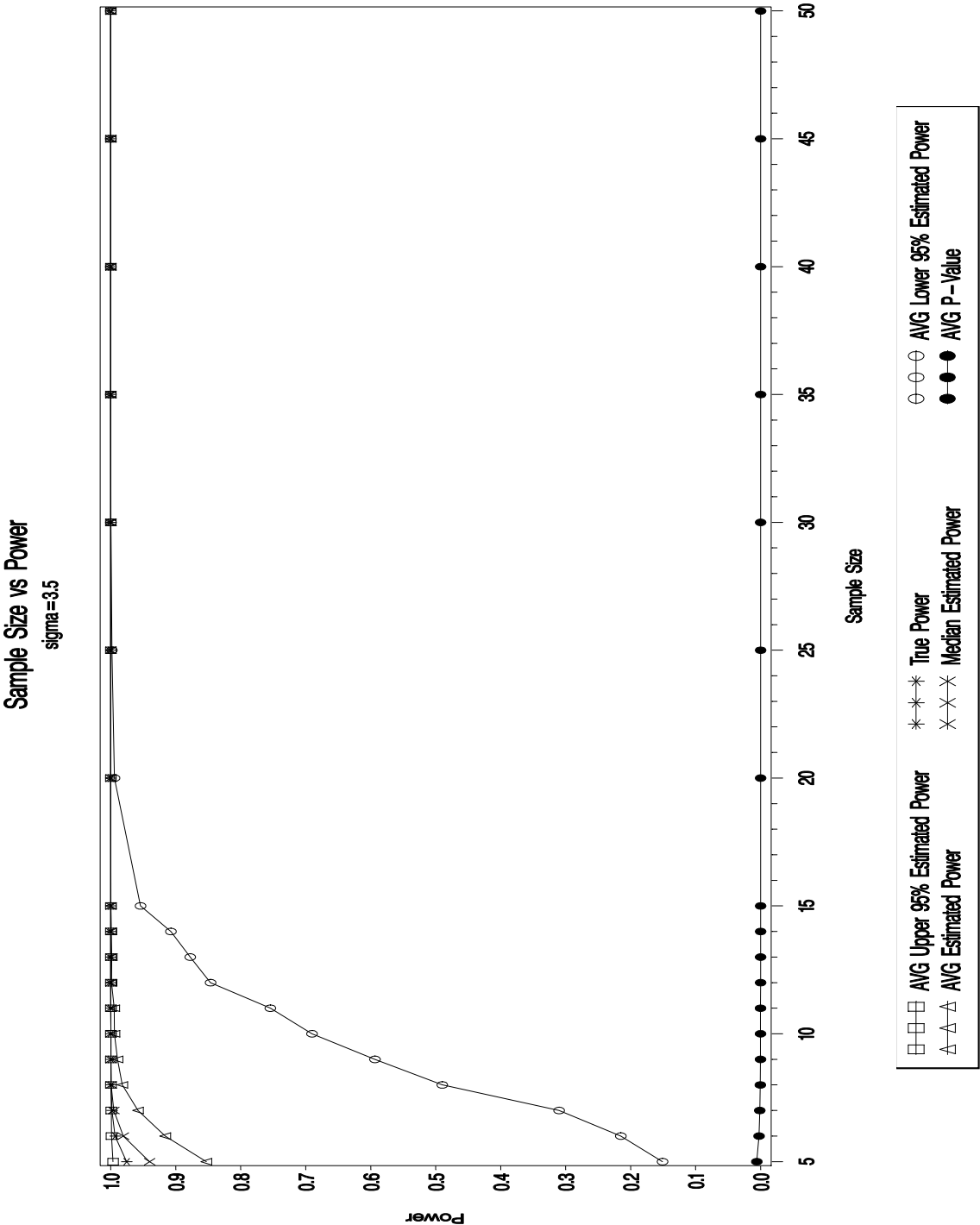
48      nr = nrow(mis);
49
50
51      /* Number of missing values */
52
53      /* Note 2 *****/
54      /*
55      /* If 'nr' is greater than
56      /* one then we have missing
57      /* values. These values are
58      /* removed from the column.
59      /*
60      *****/
61
62      /* Remove initialize position */
63      /* Transpose a column to row */
64      /* Remove missing values */
65      /* Transpose row to a column */
66
67      /* Find median */
68
69
70      if nr>1 then mis = remove(mis,1);
71      if nr>1 then col = col`;
72      if nr>1 then col = remove(col,mis);
73      if nr>1 then col = col`;
74
75      med = median(col);
76
77      /* Concatinate median to average row */
78
79      avgRows=avgRows||med;
80
81
82      create amsim&itB from avgRows;
83      append from avgRows;
84
85      quit;
86      %mend avgMedSim4;

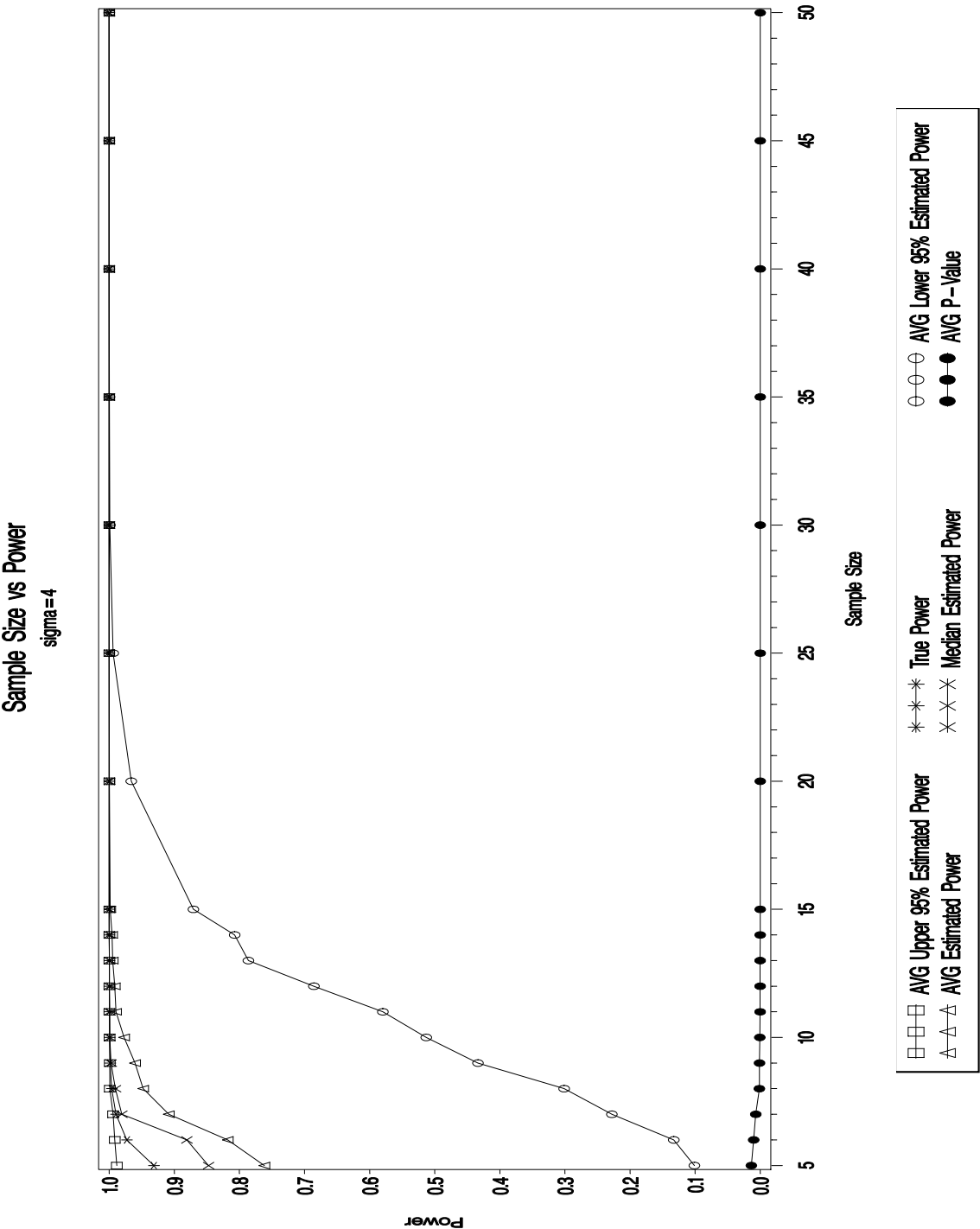
```

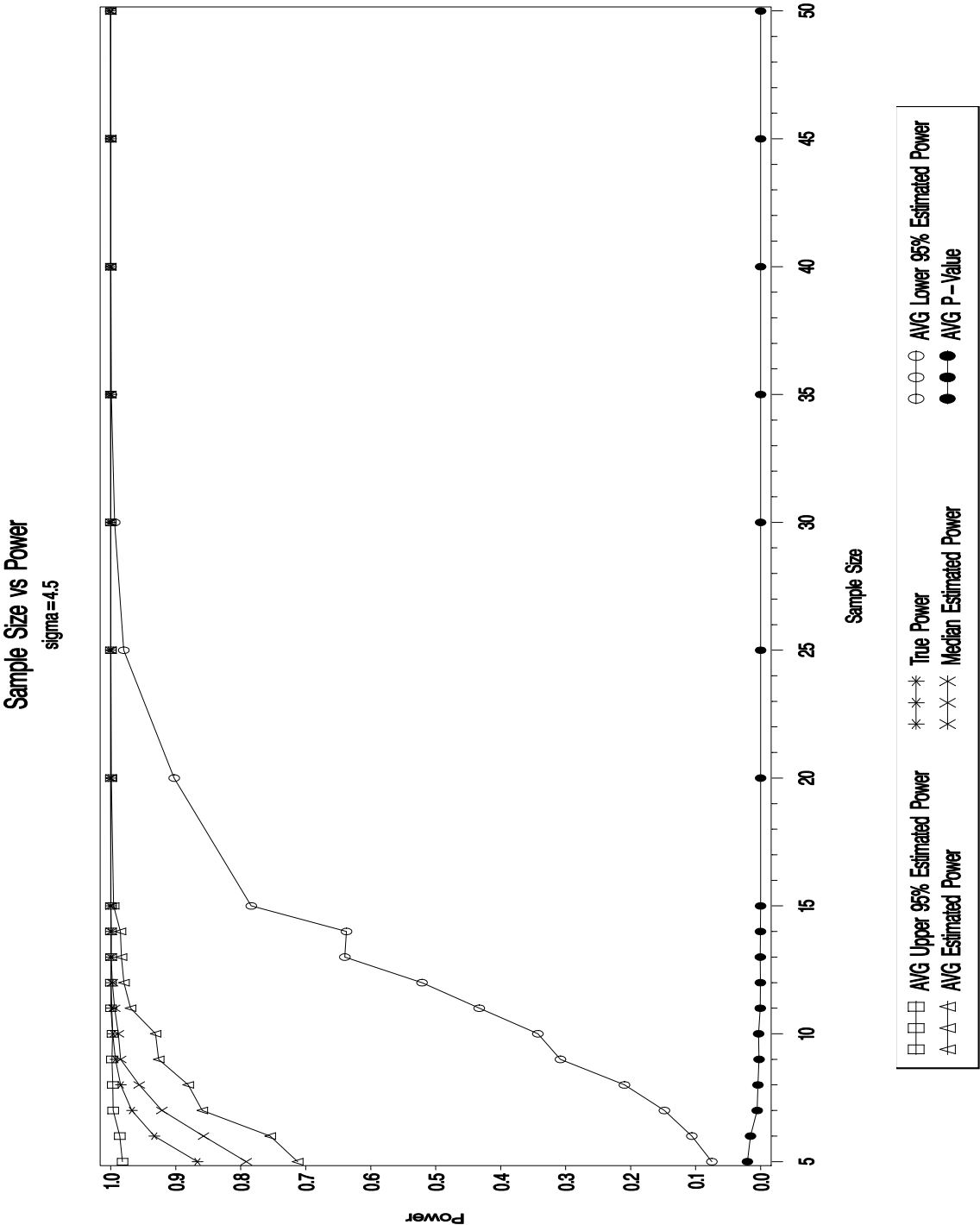


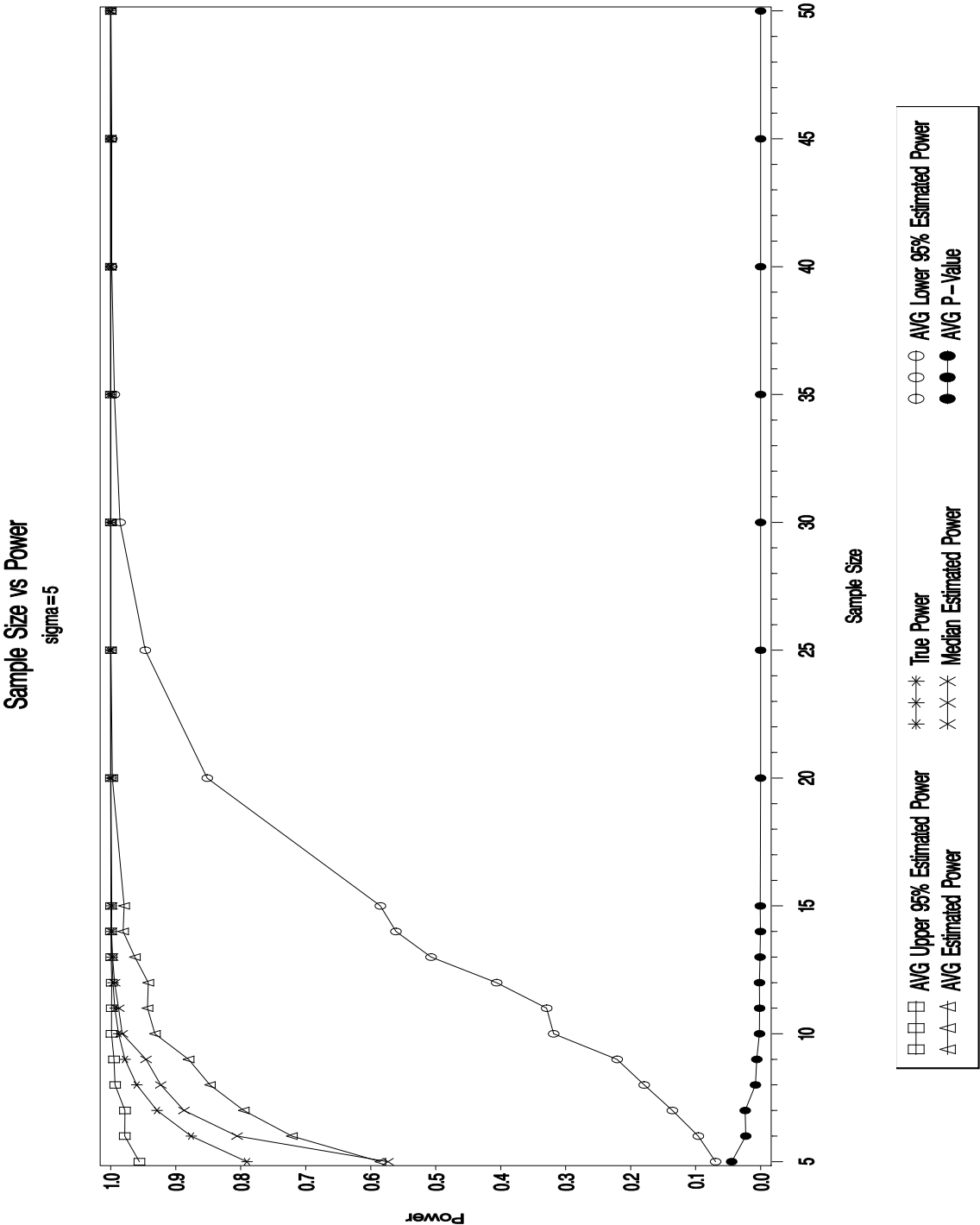


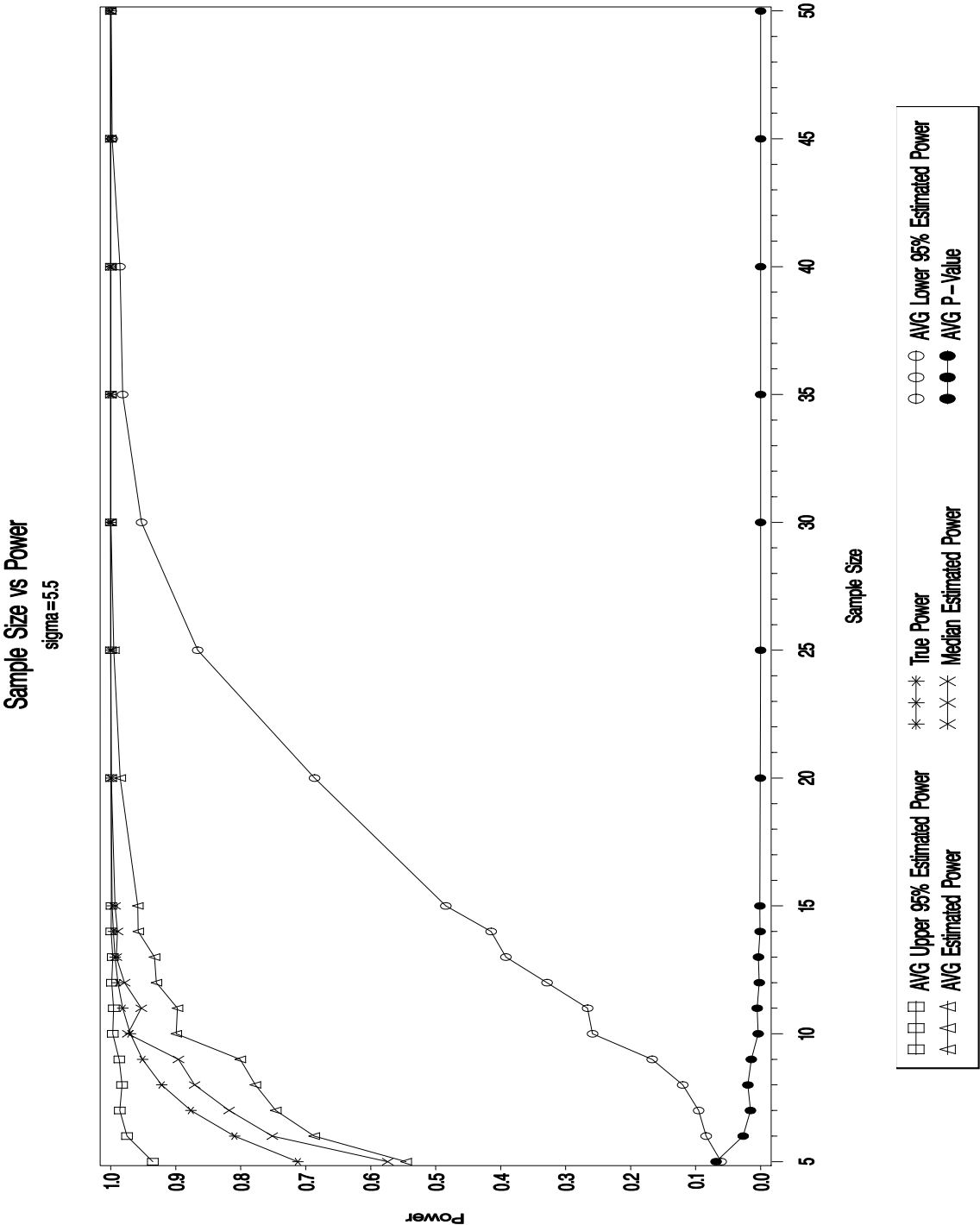


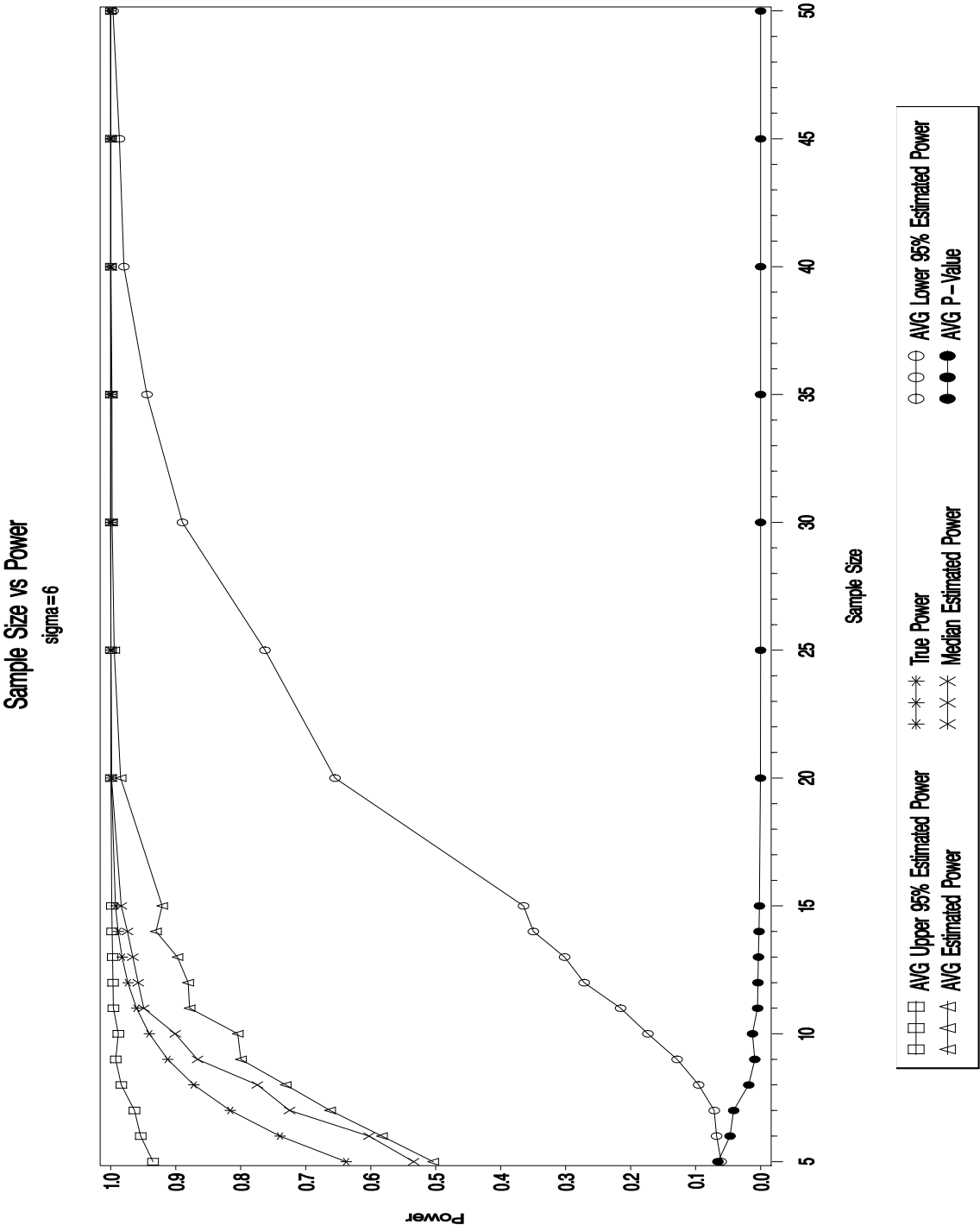




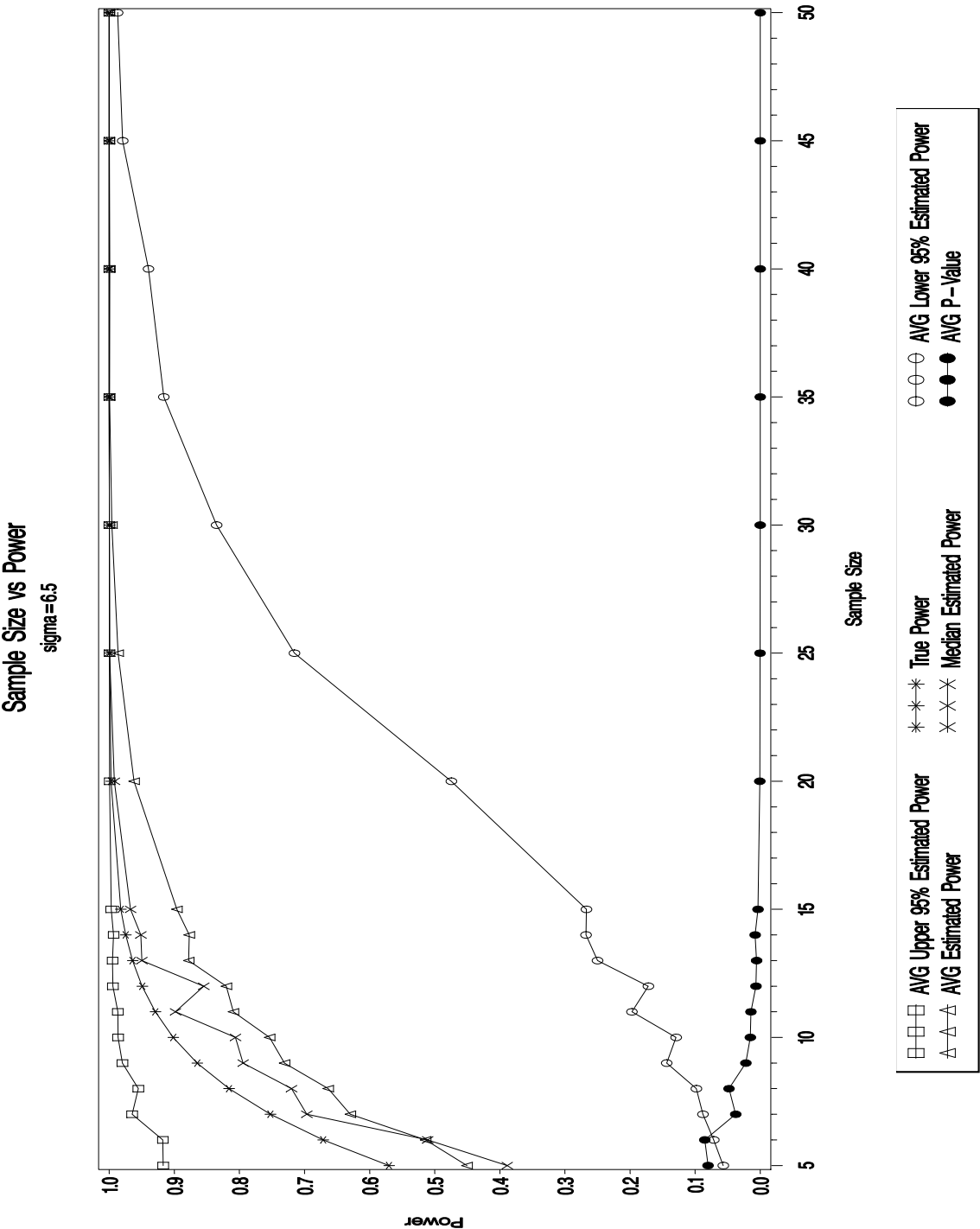


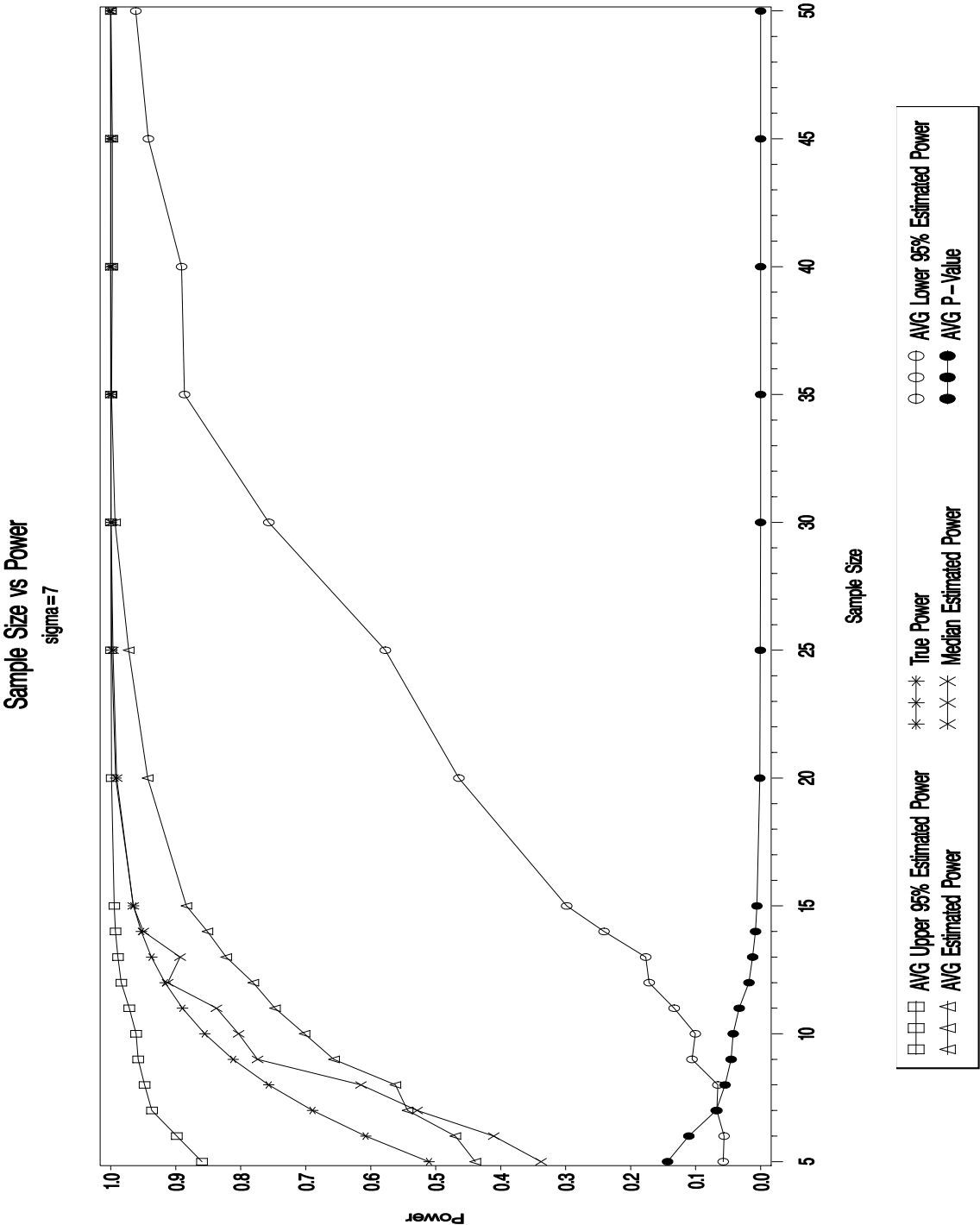


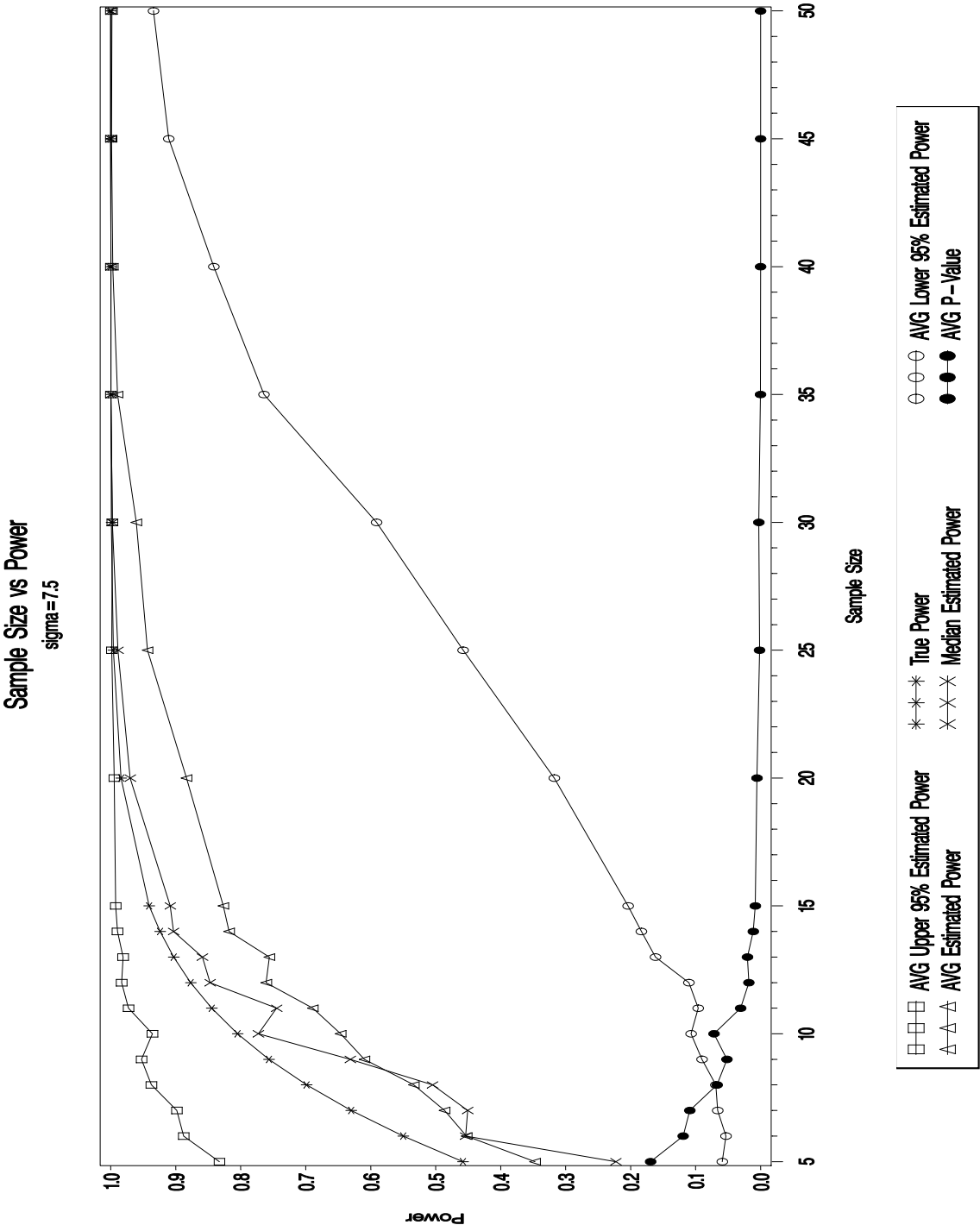


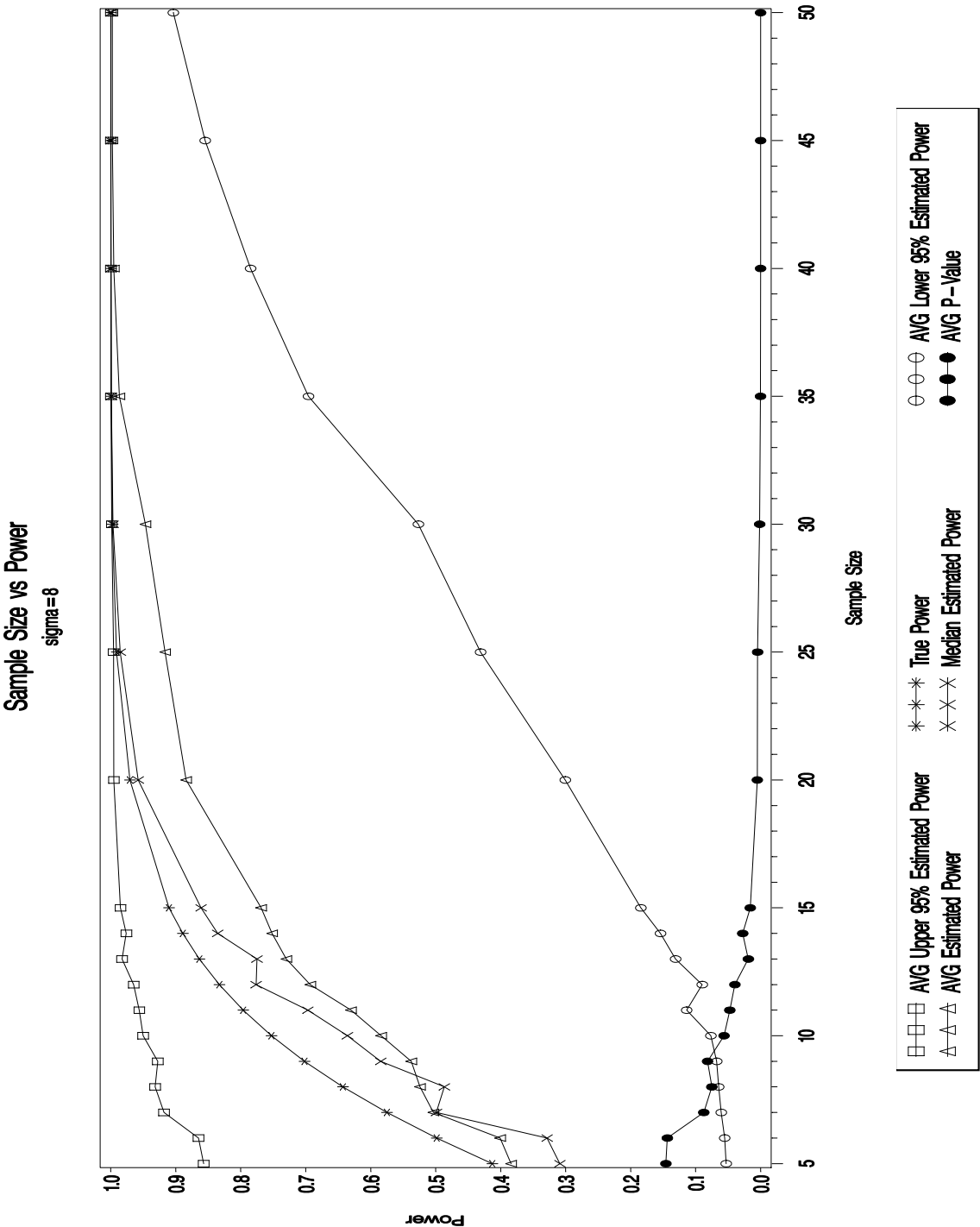


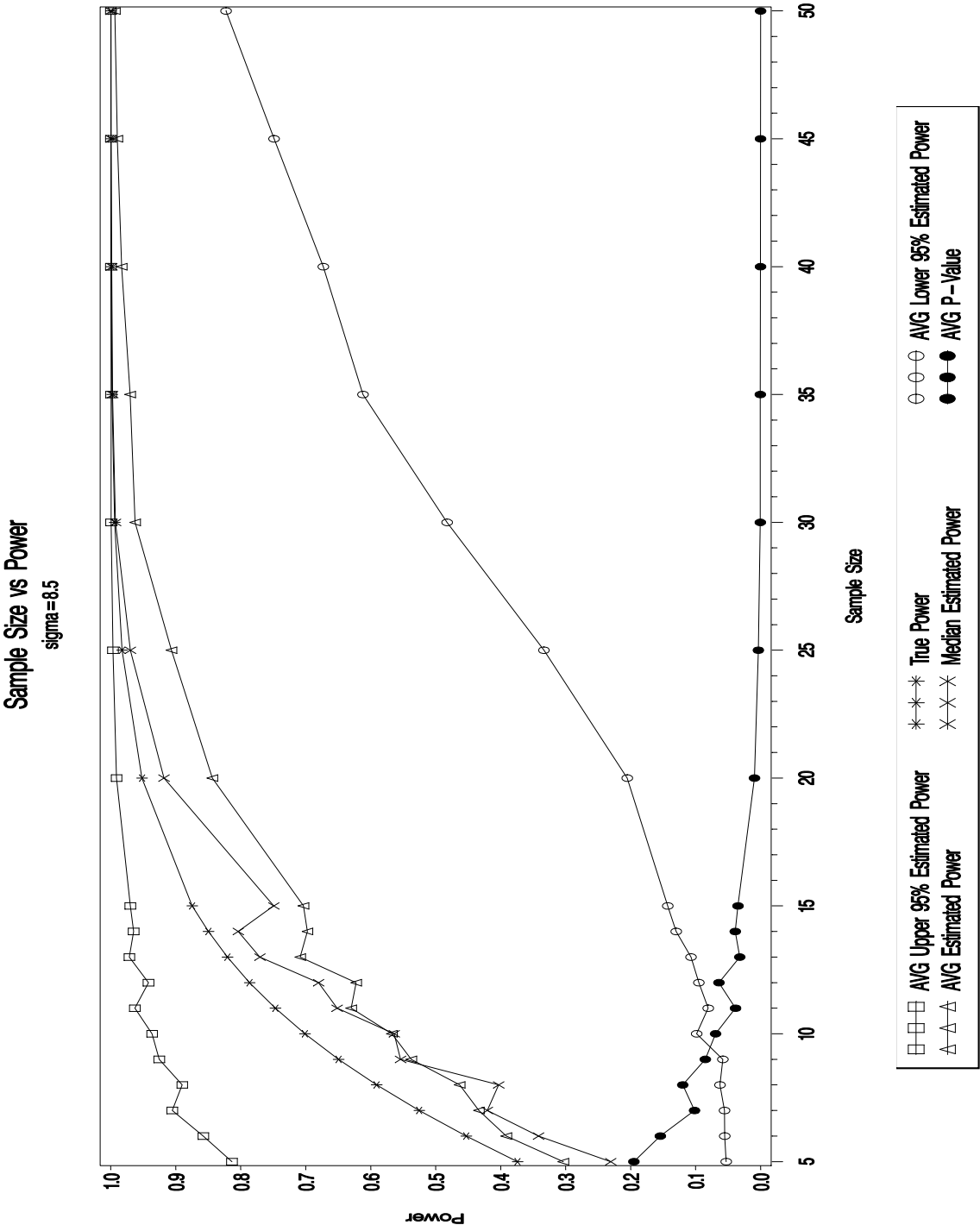


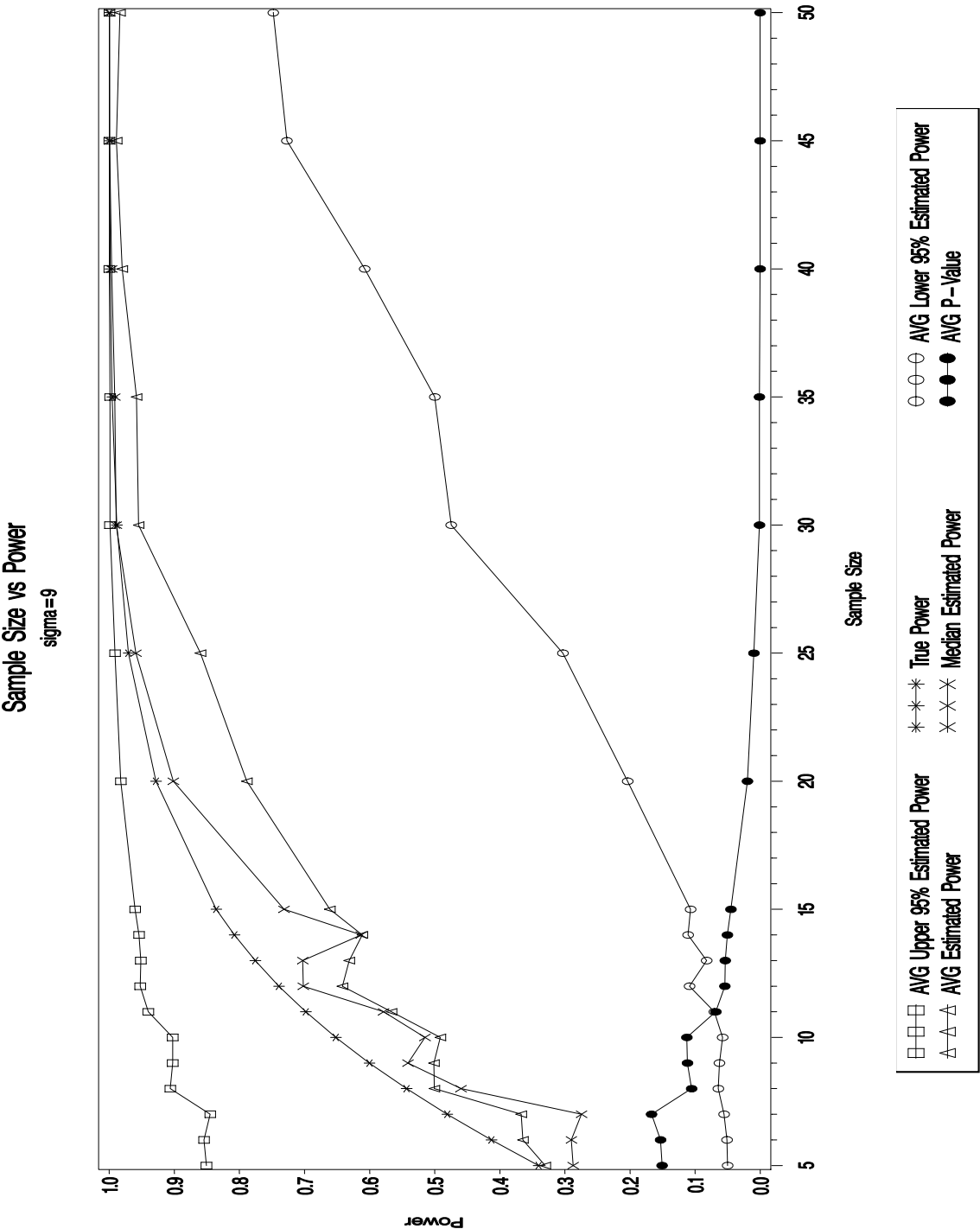


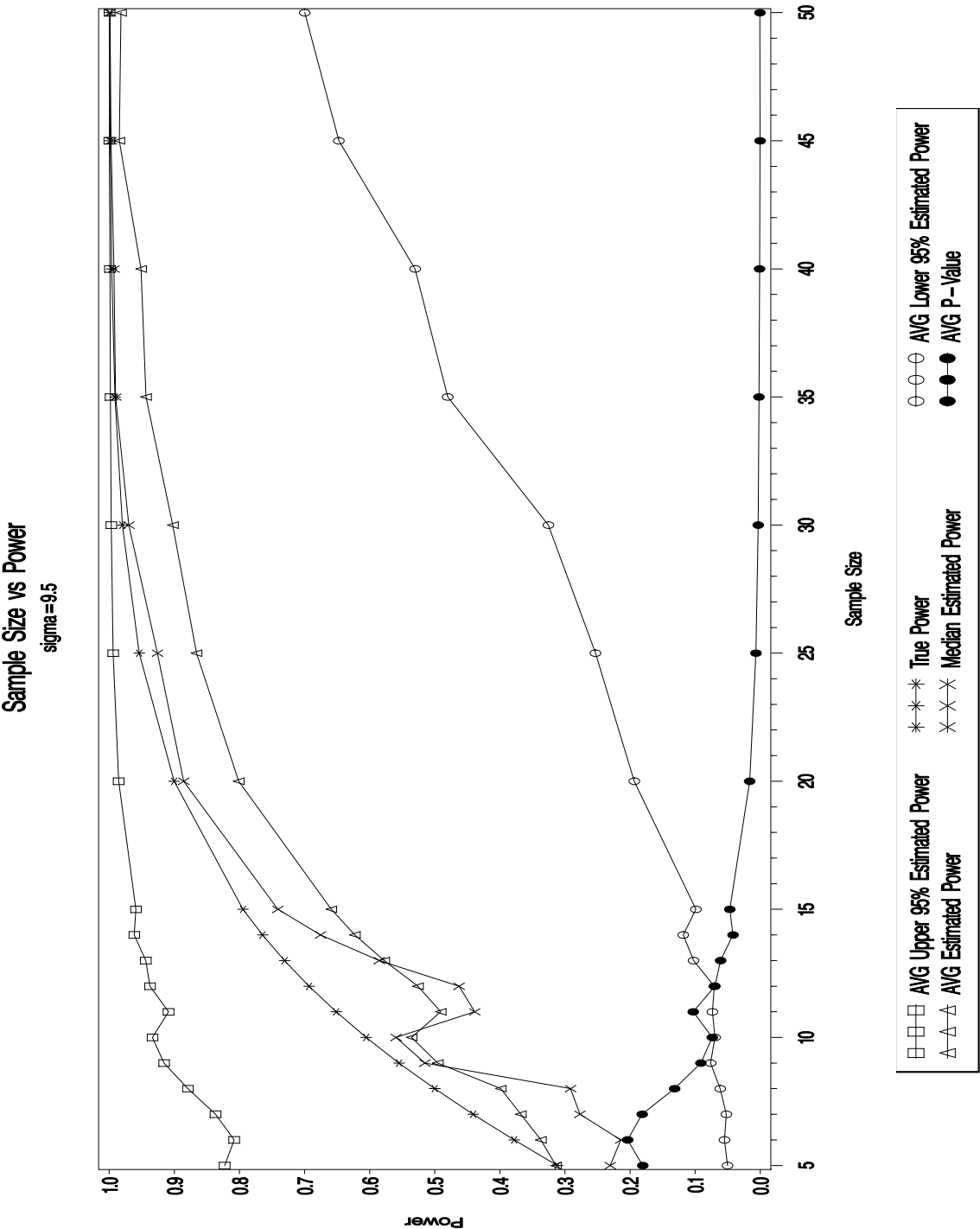


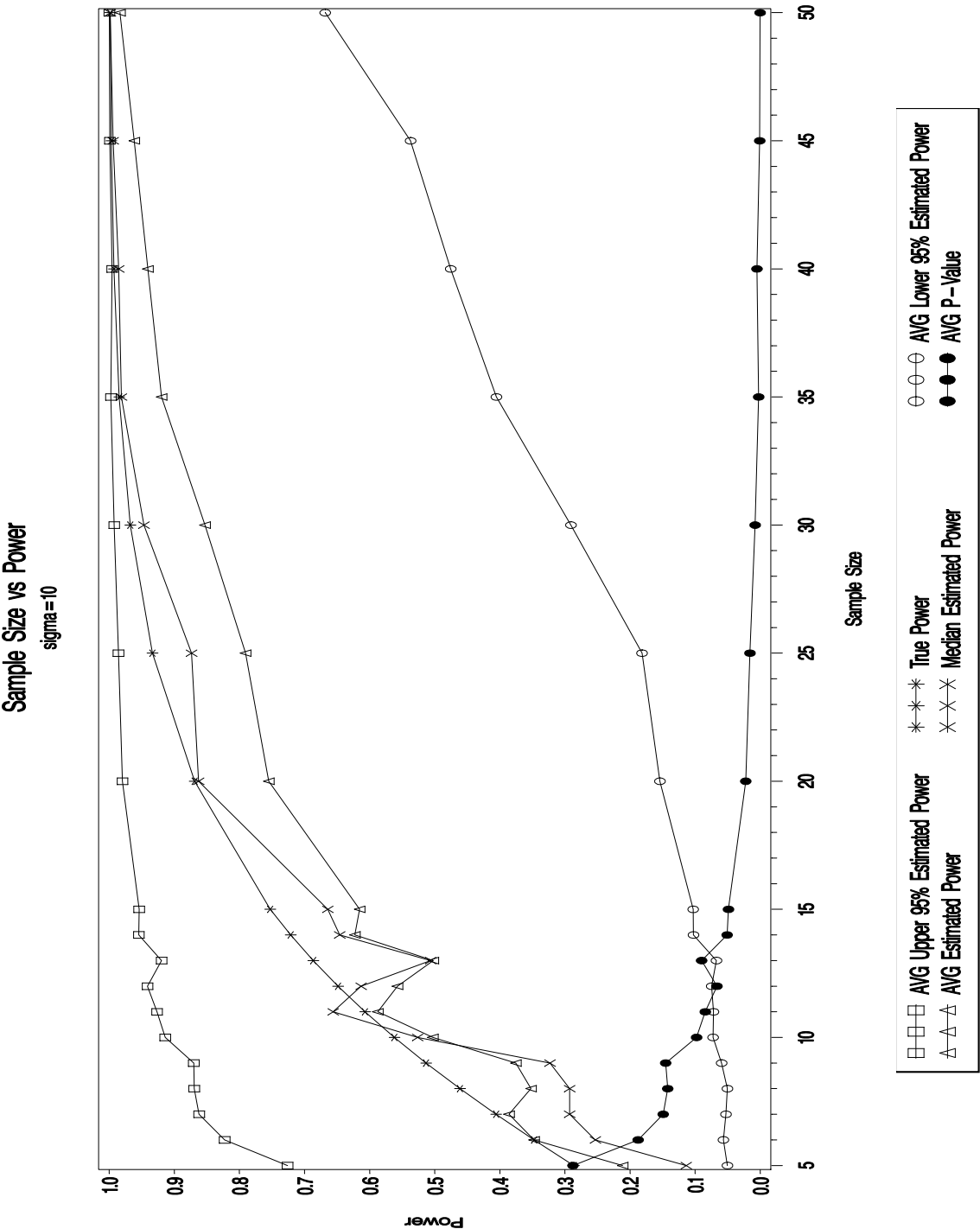




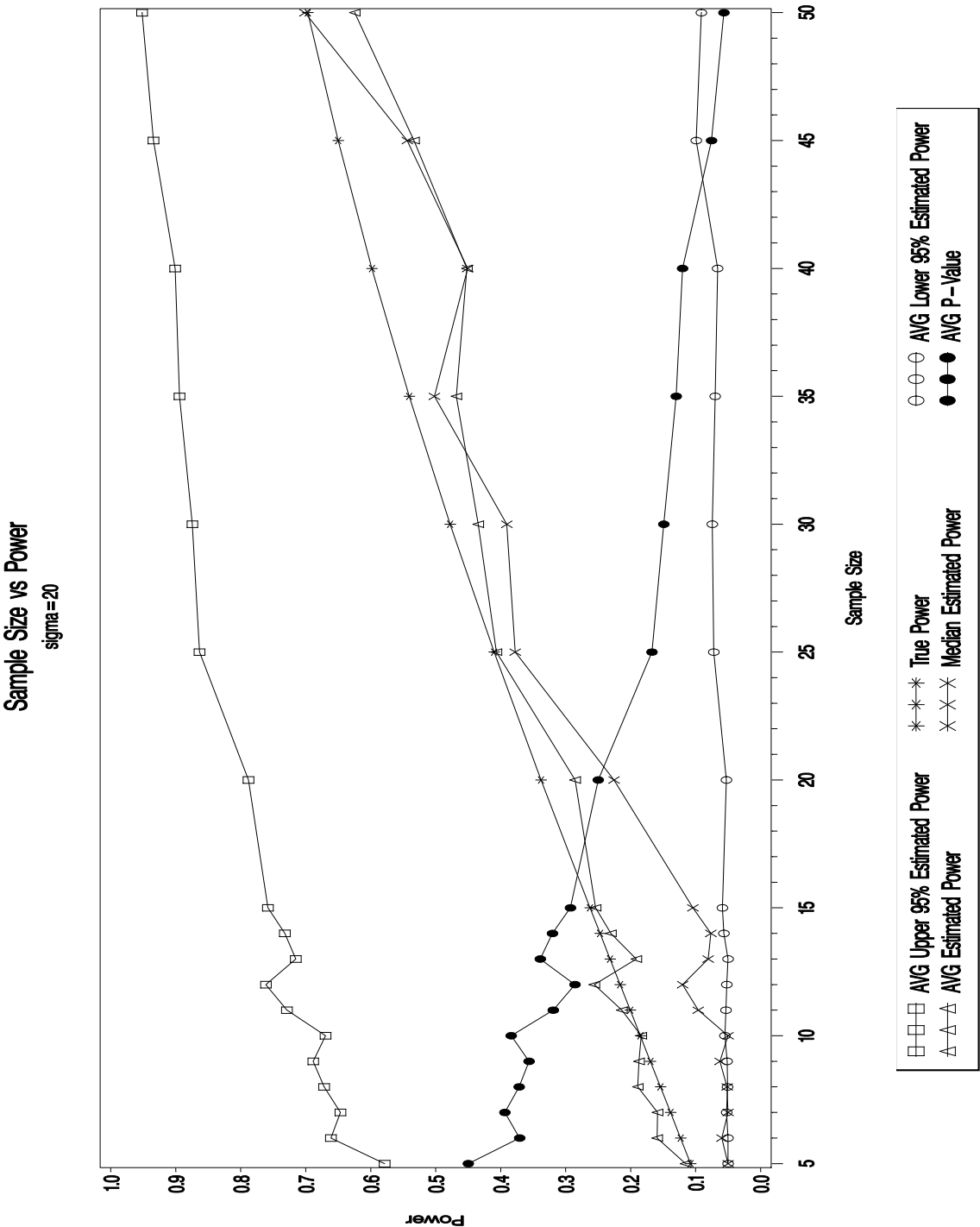


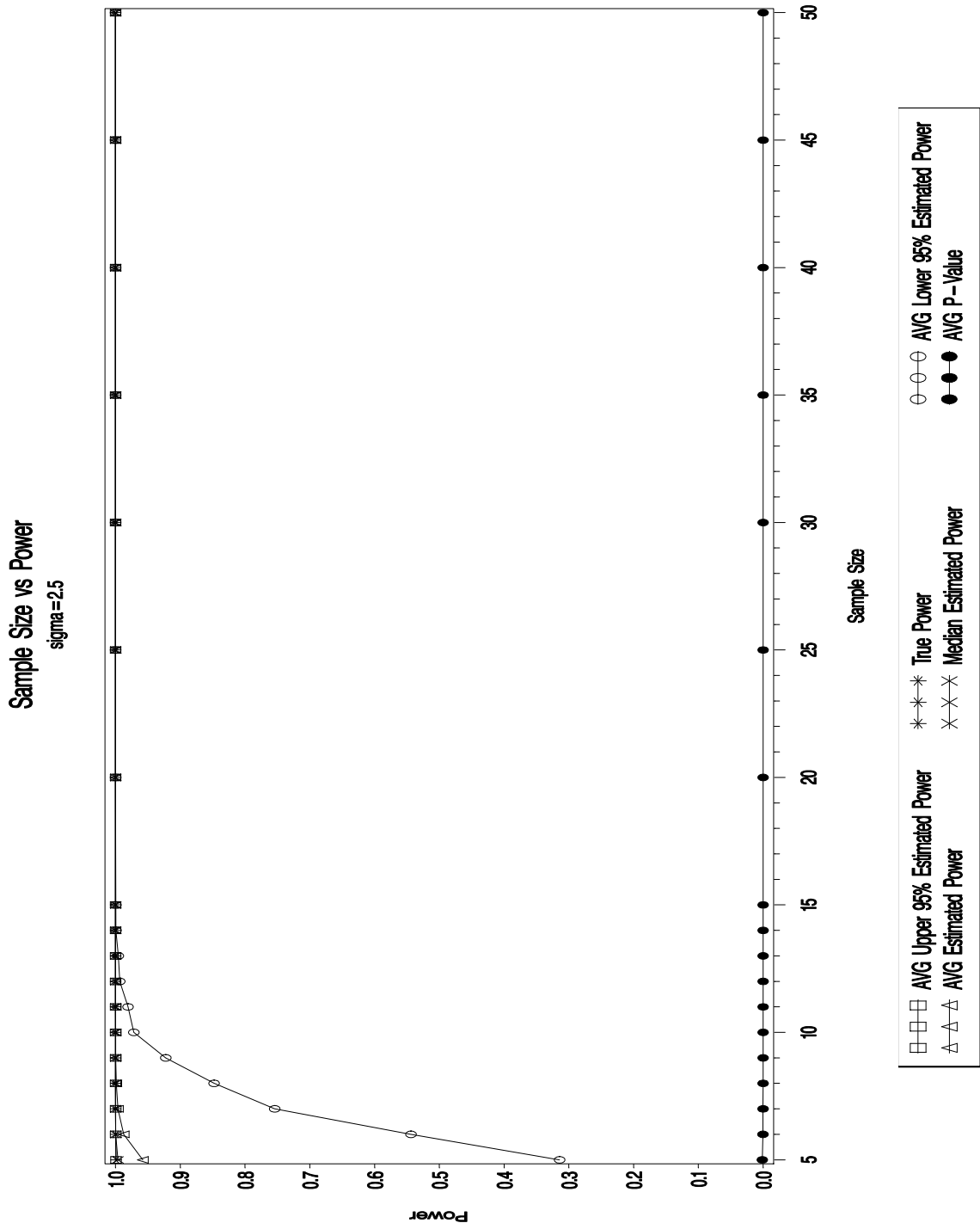


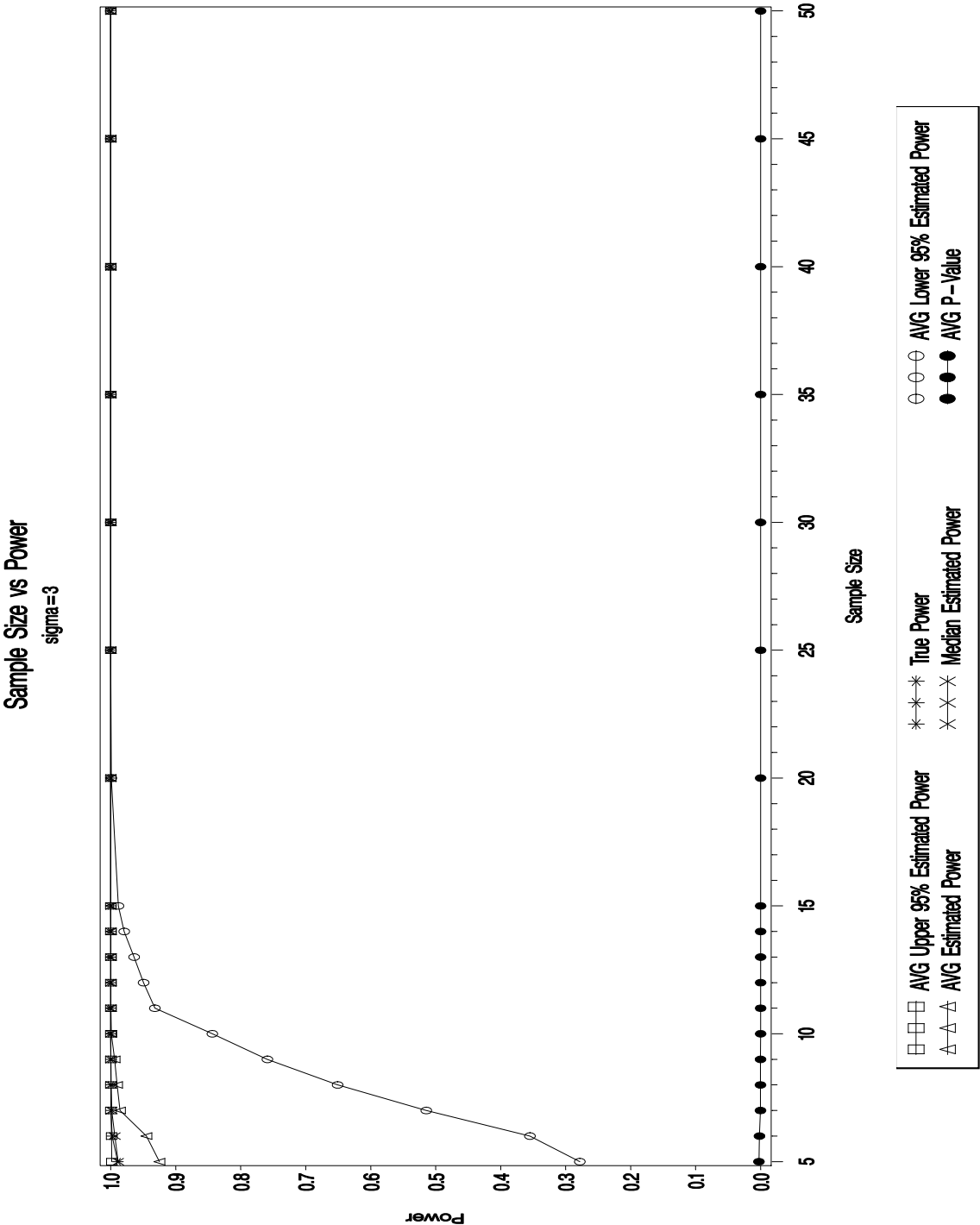


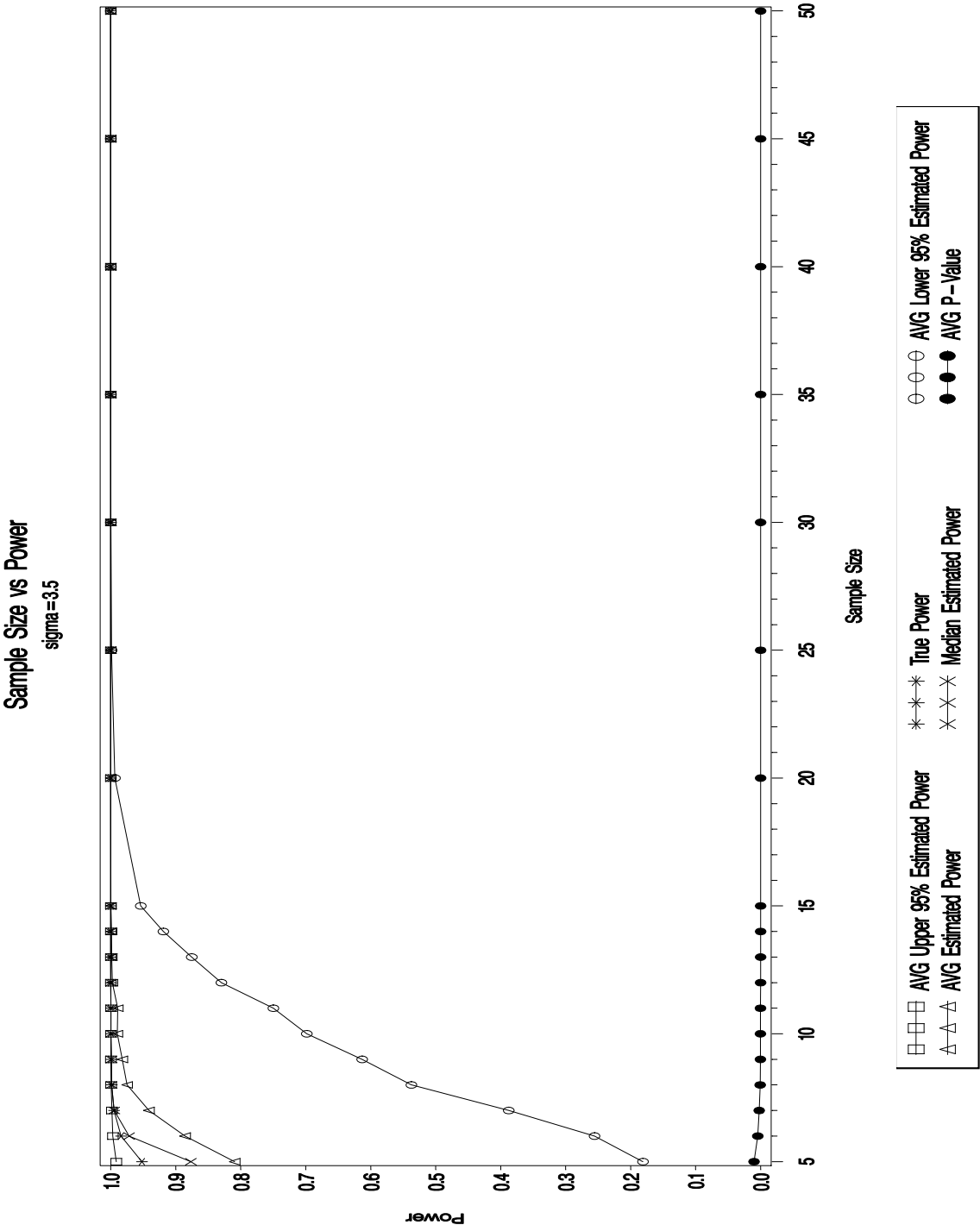


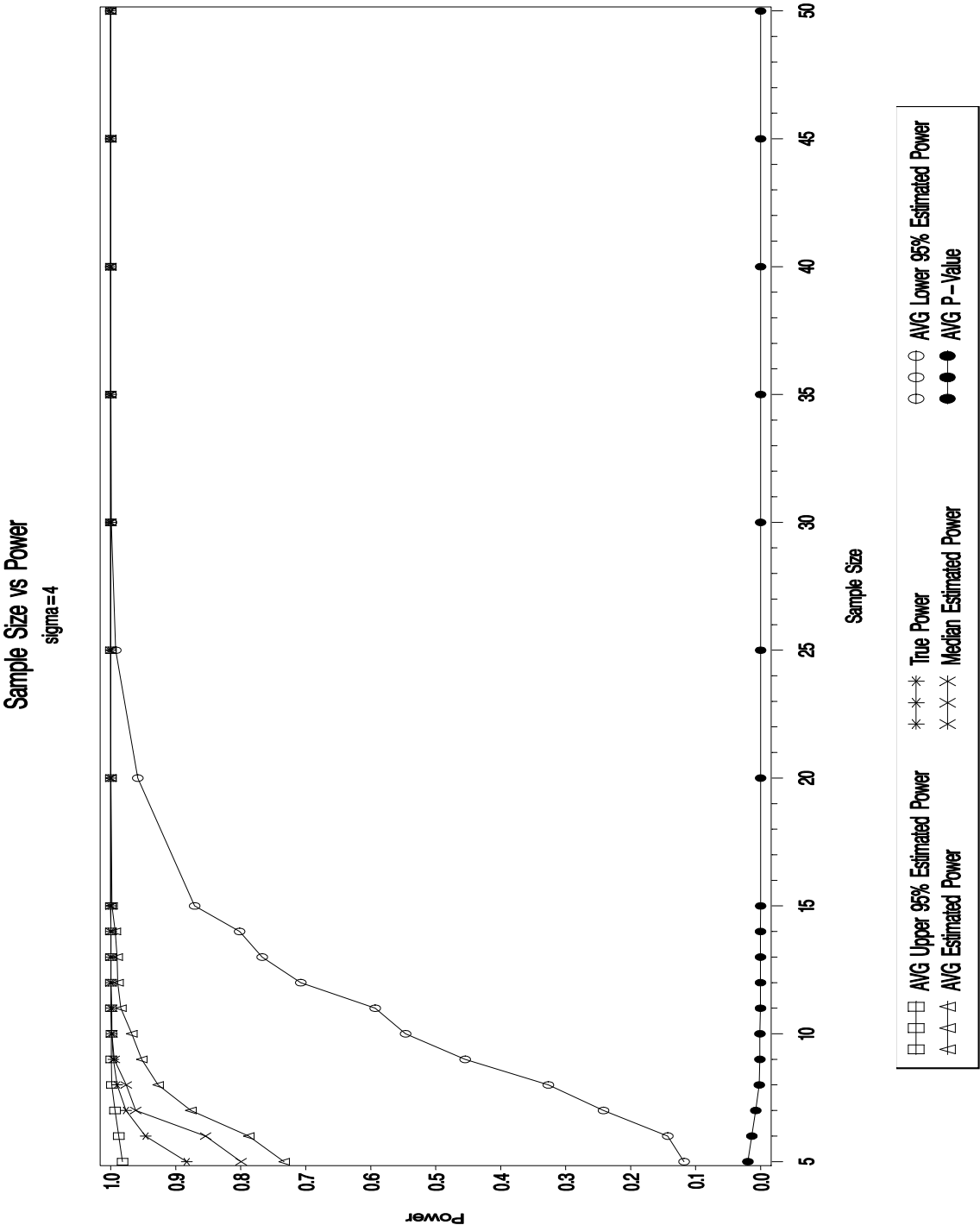


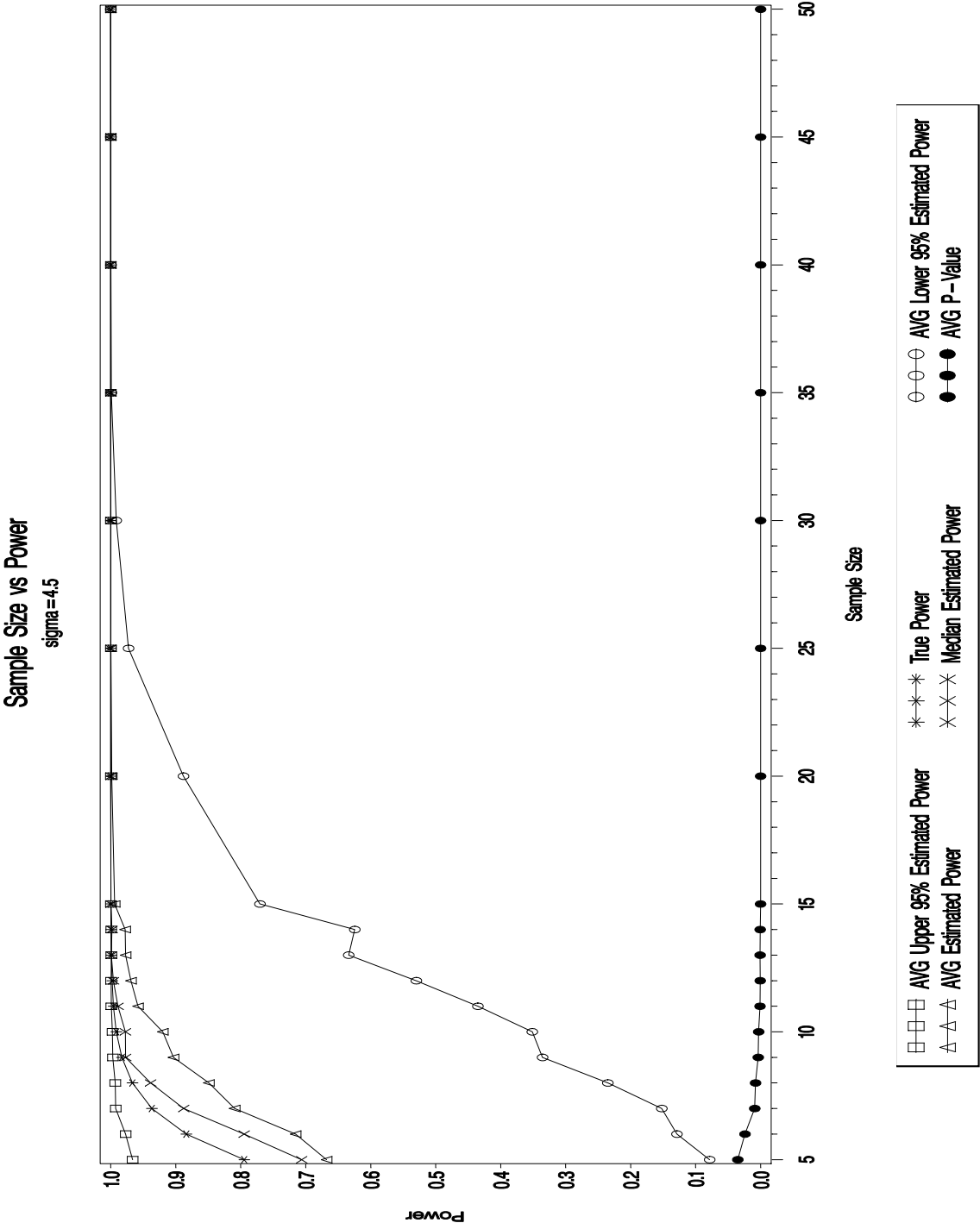


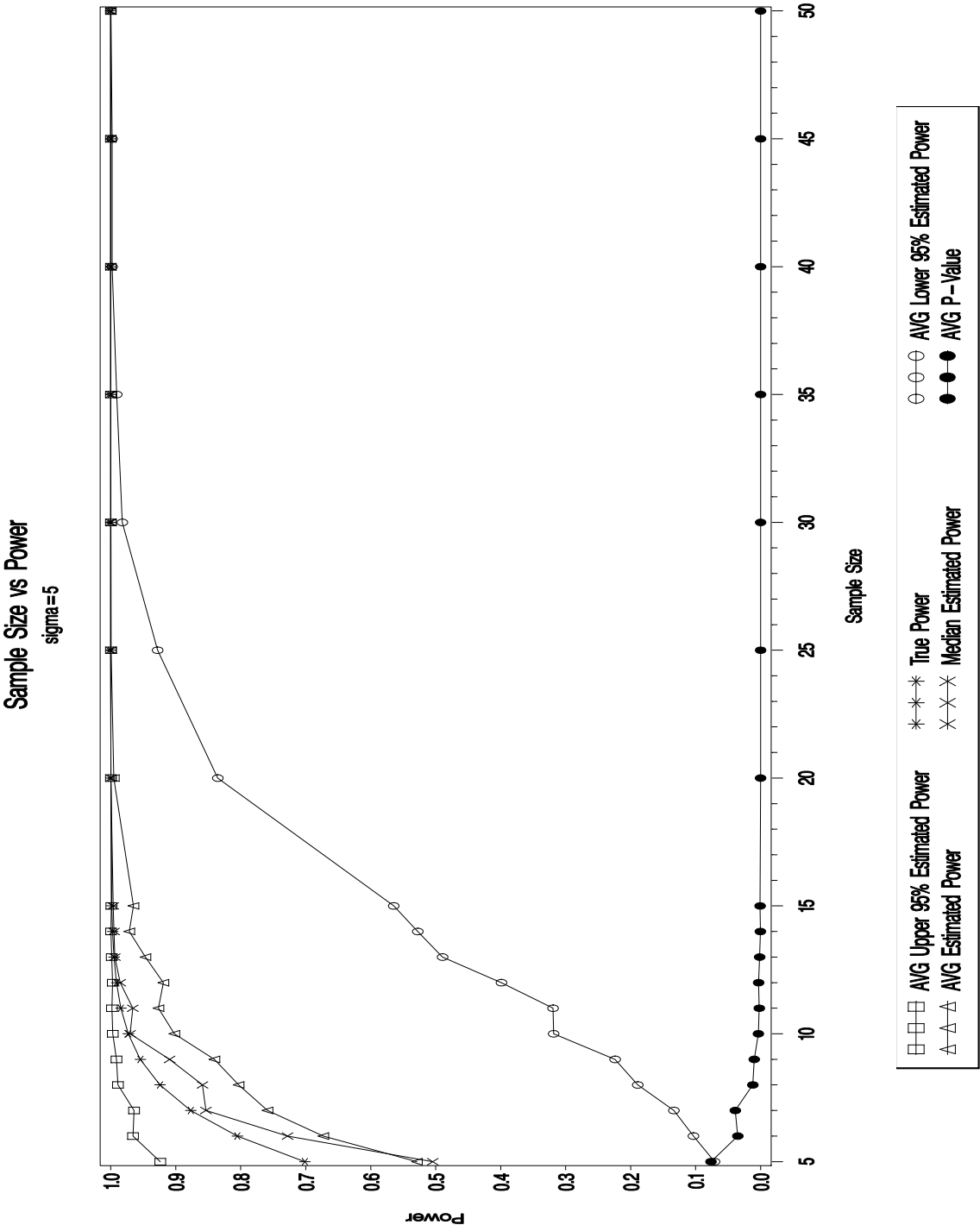


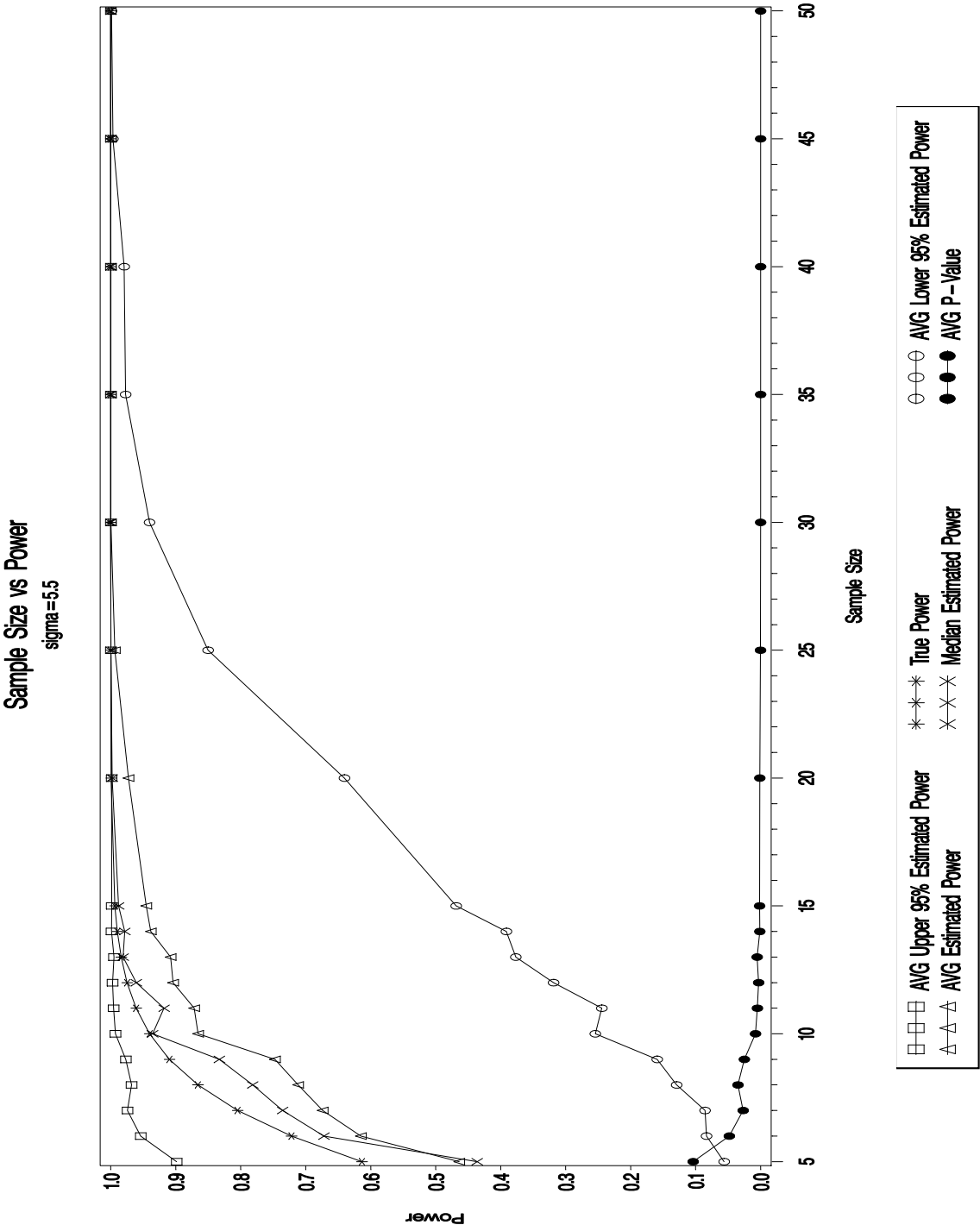




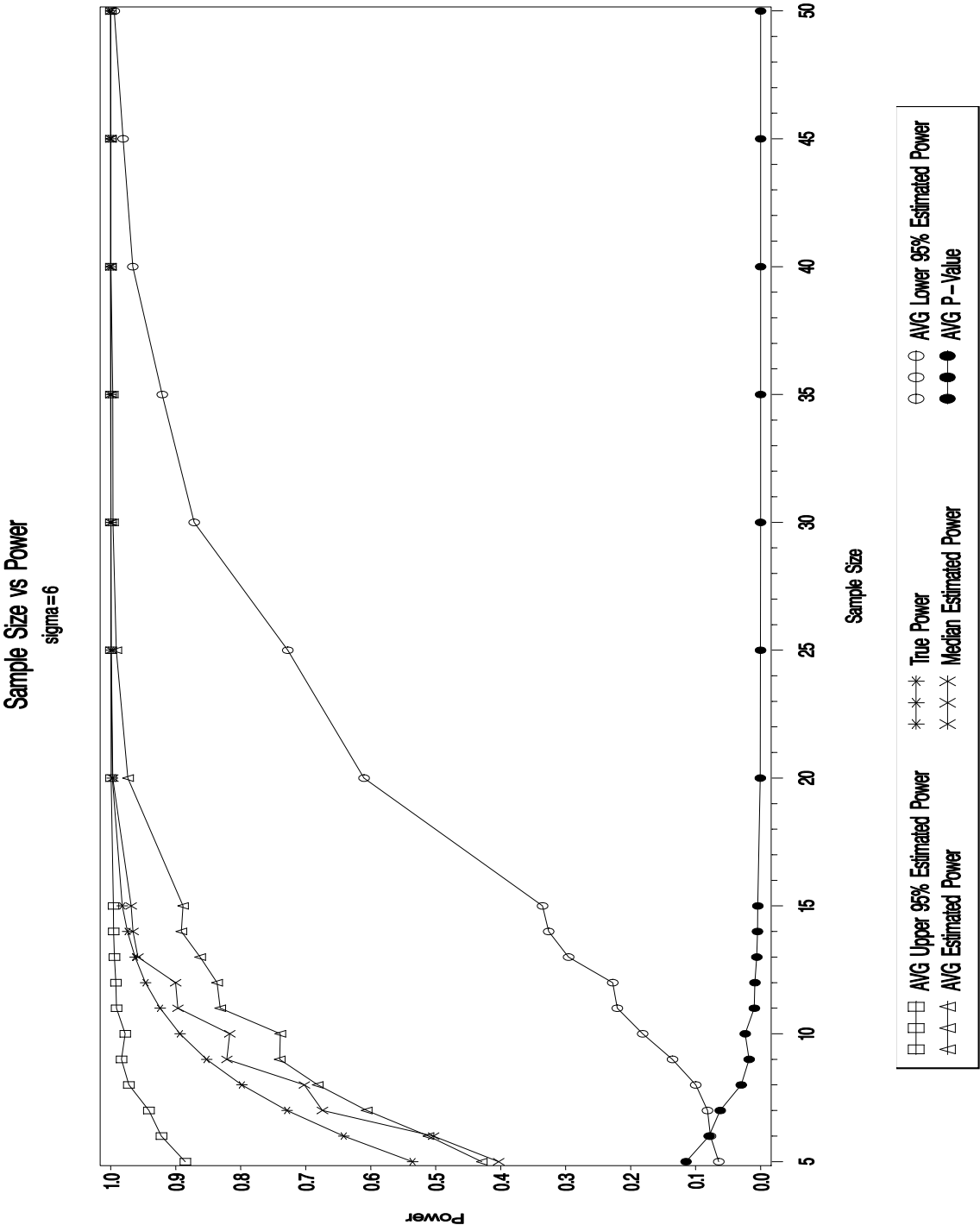


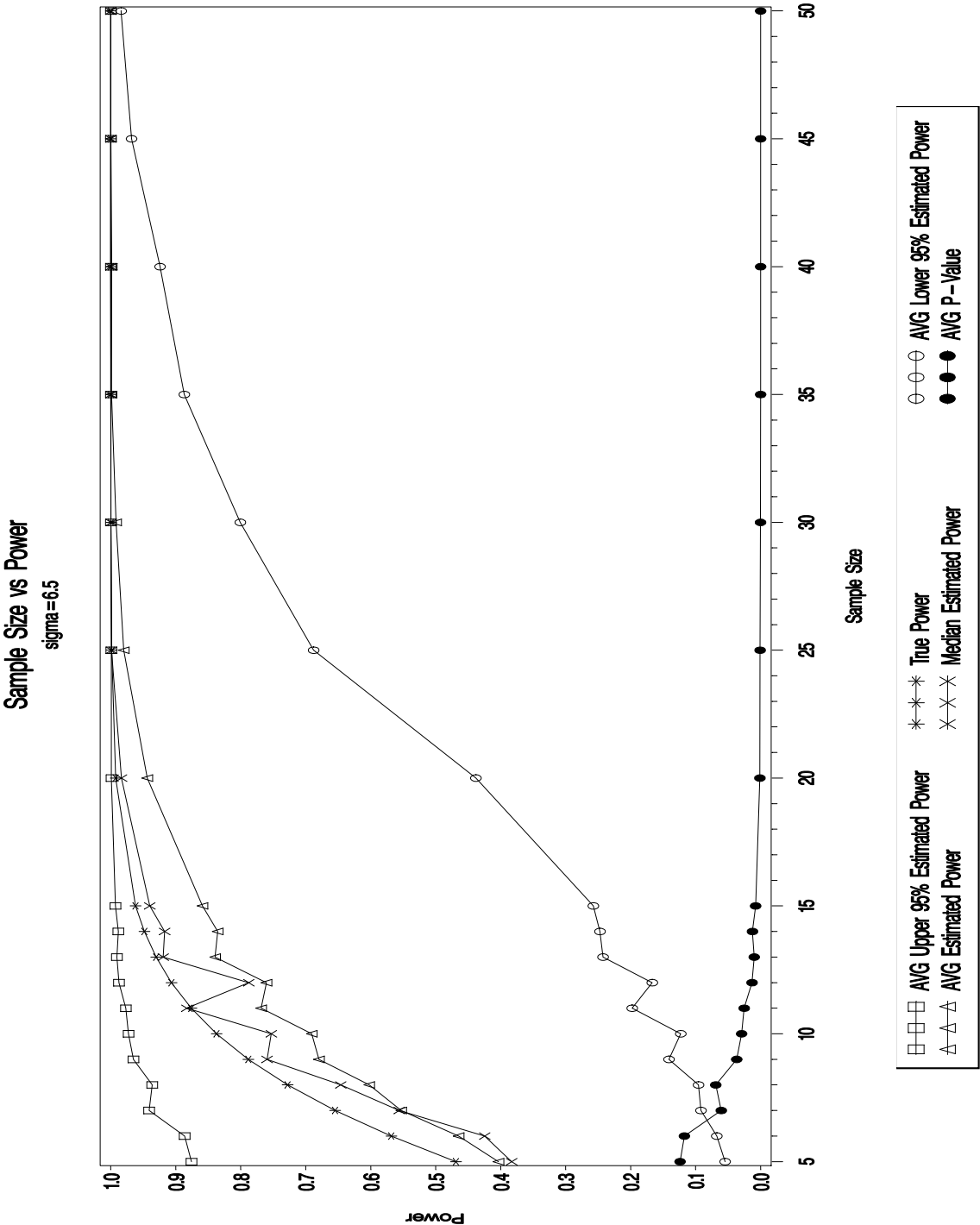


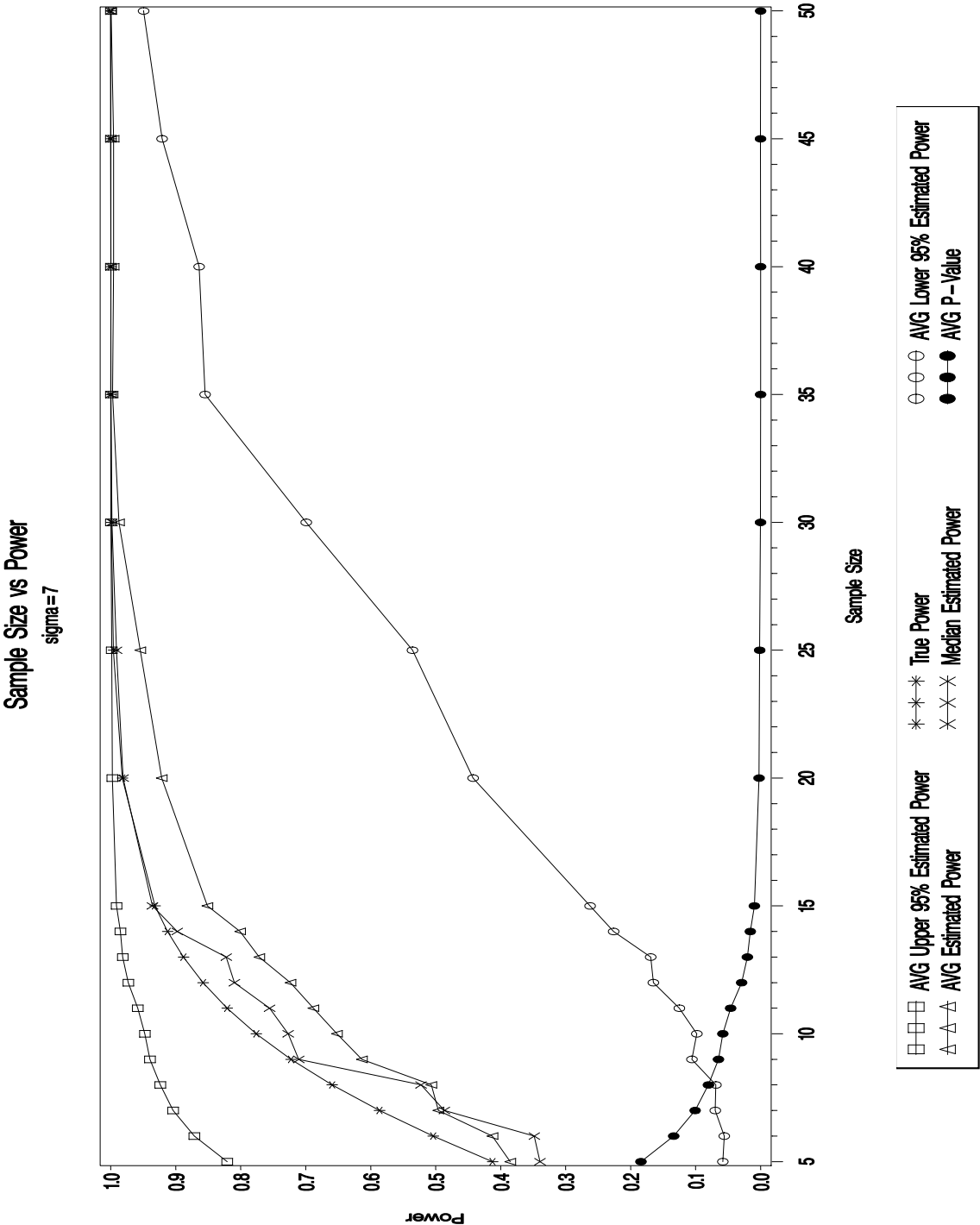


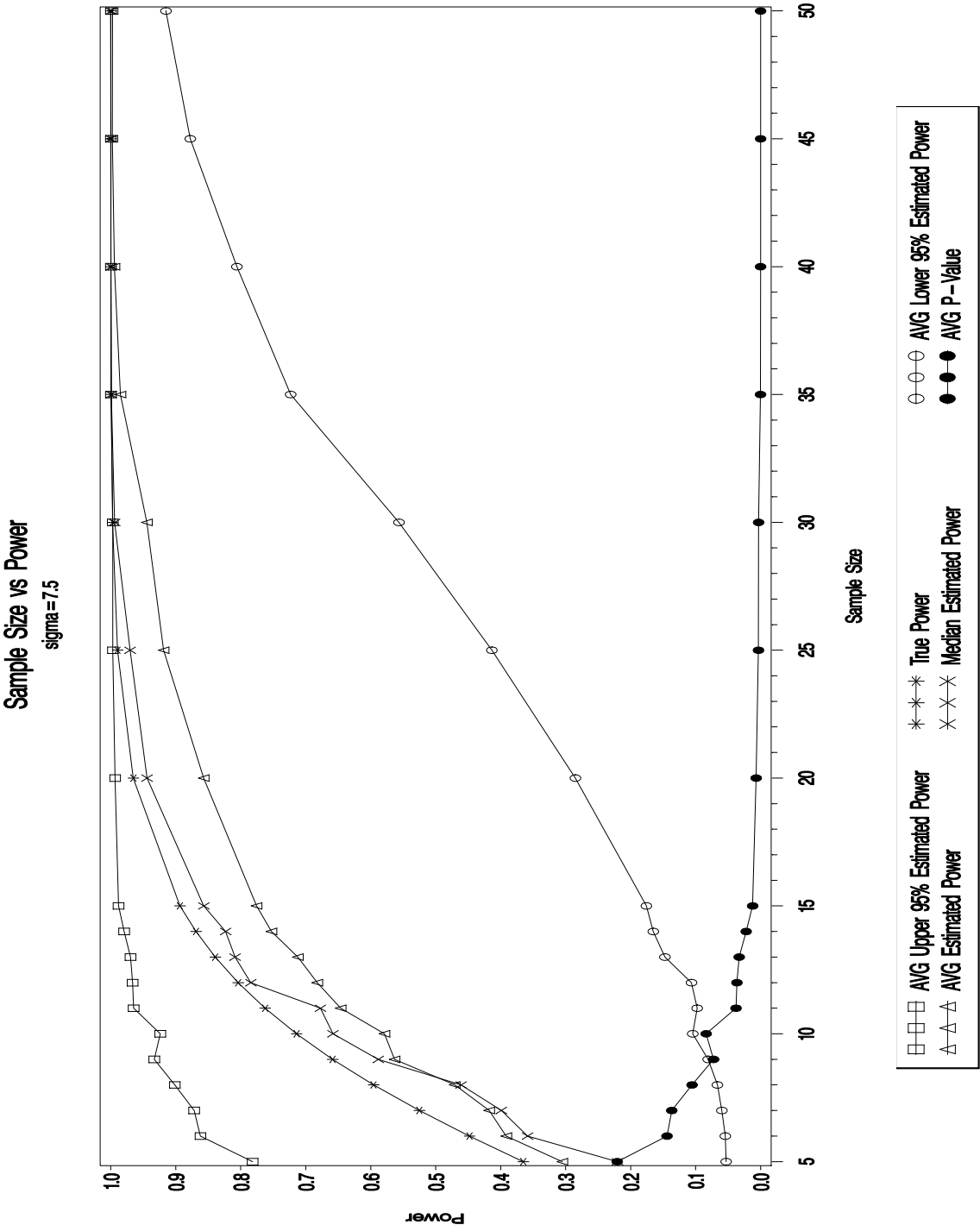


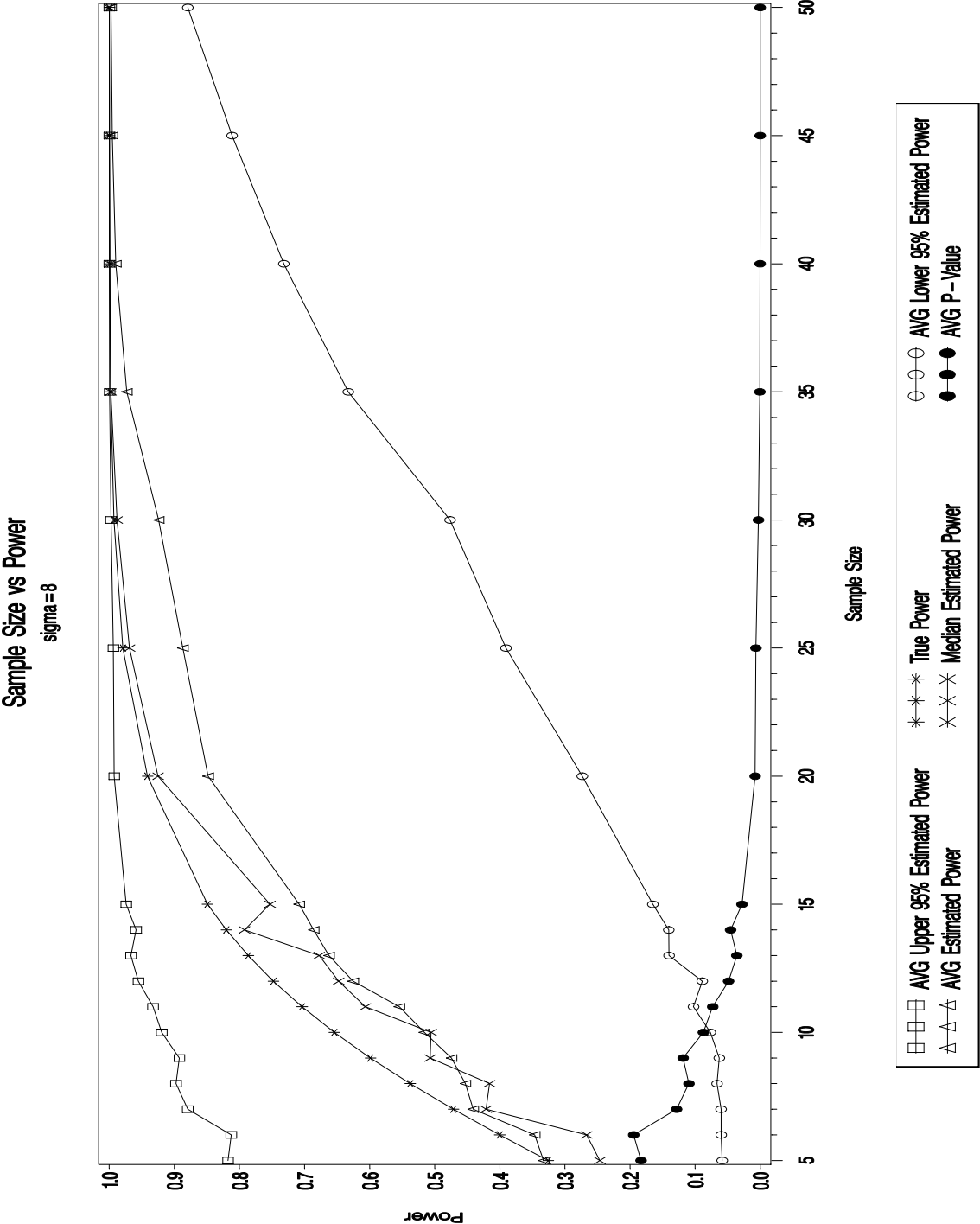


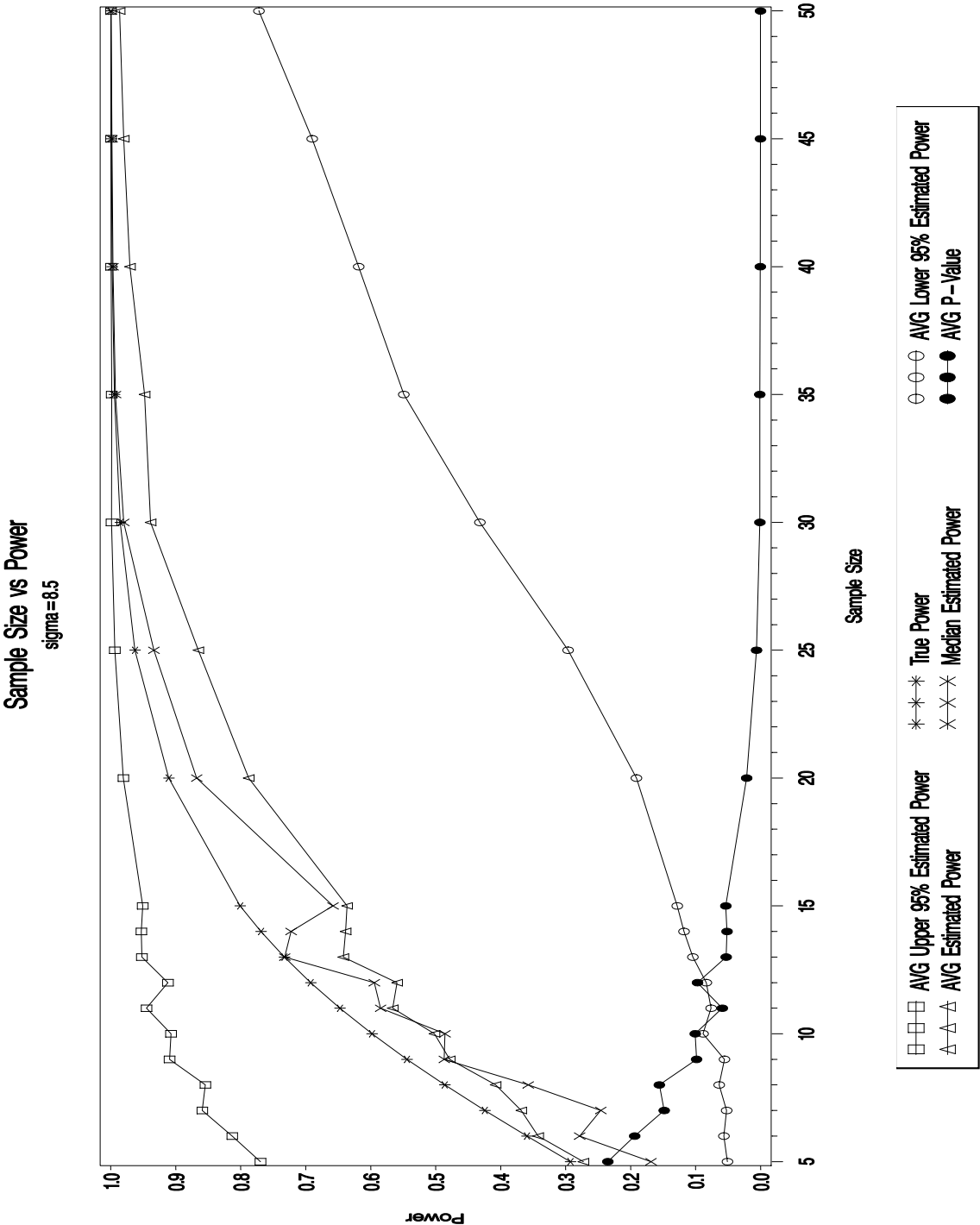


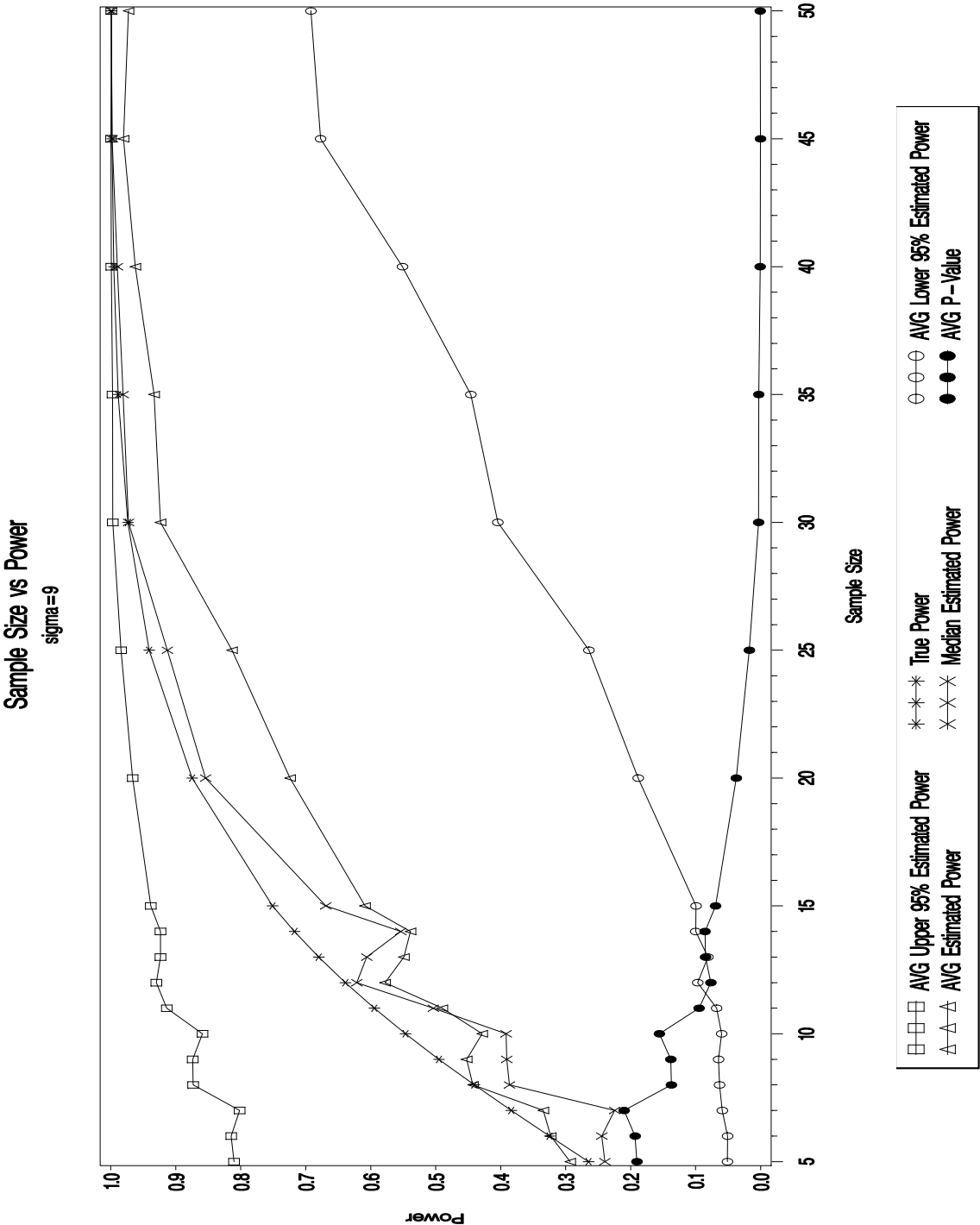


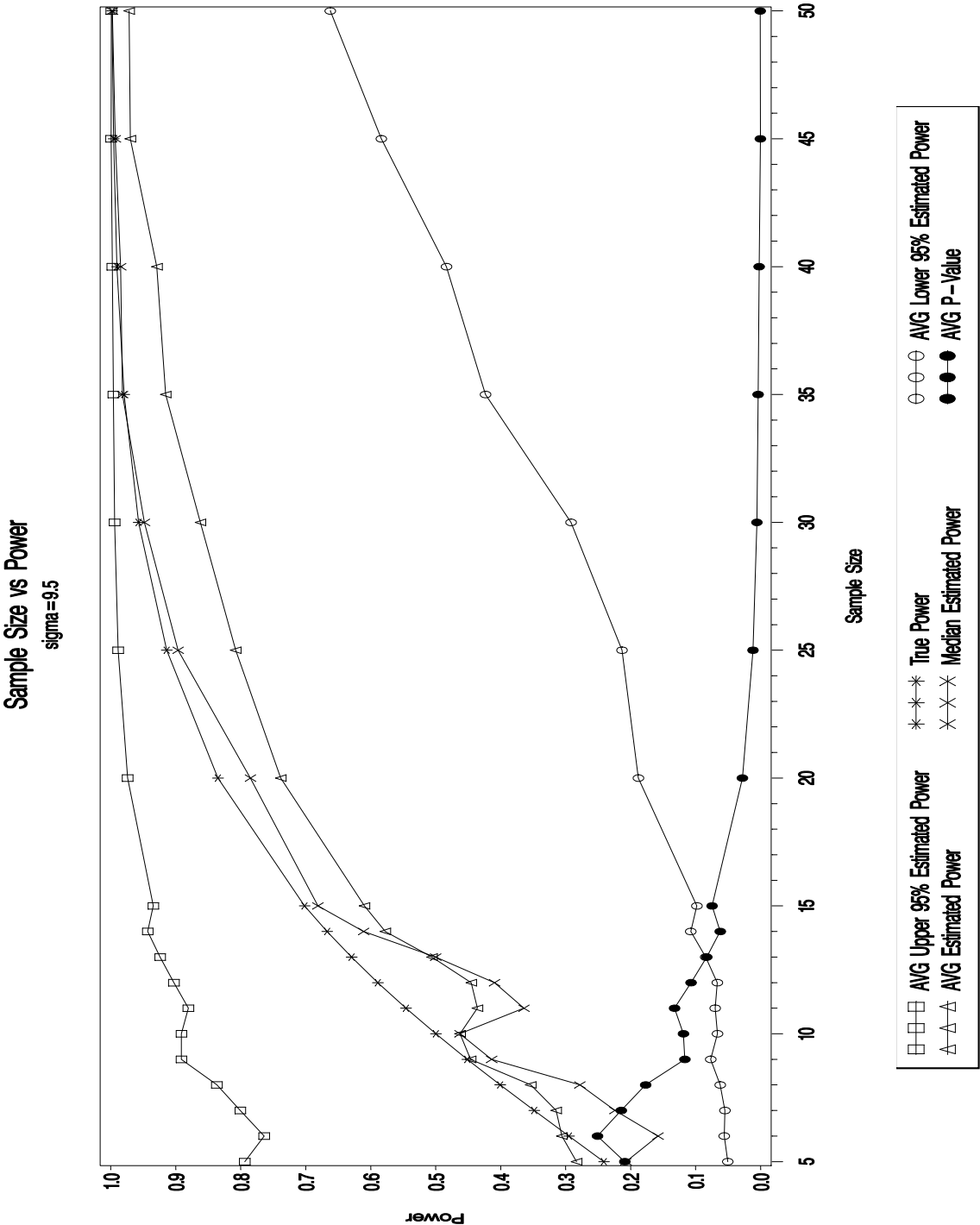




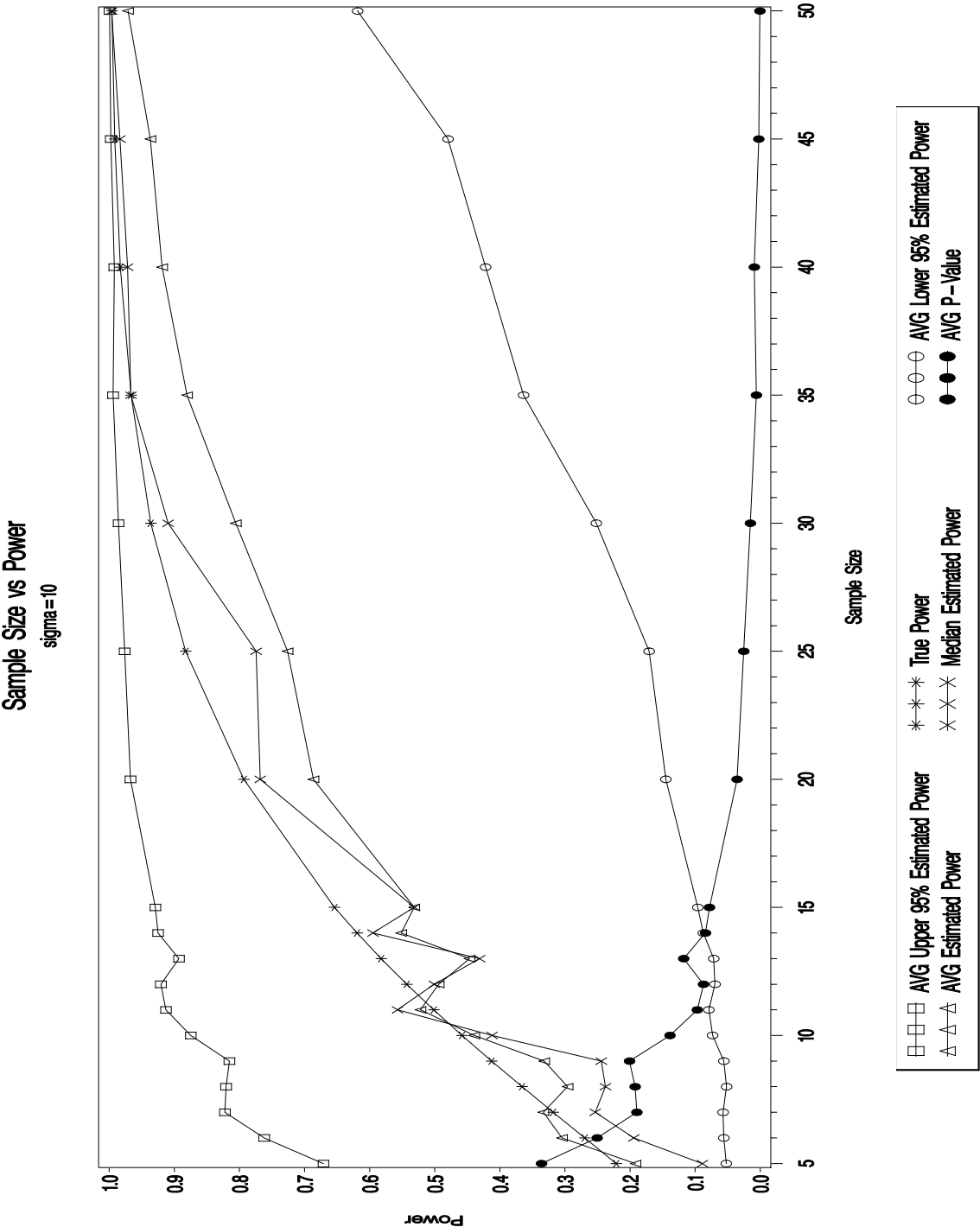


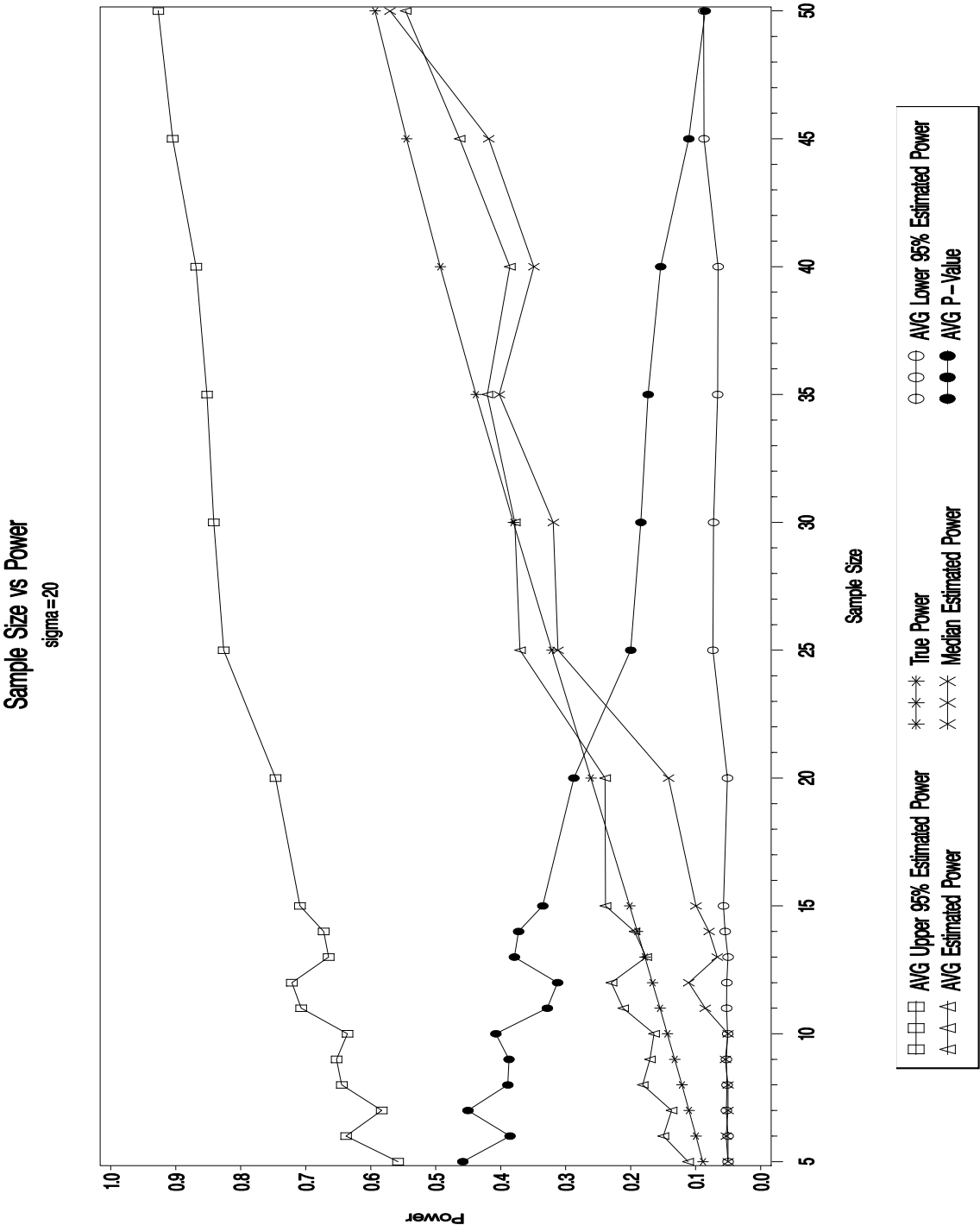


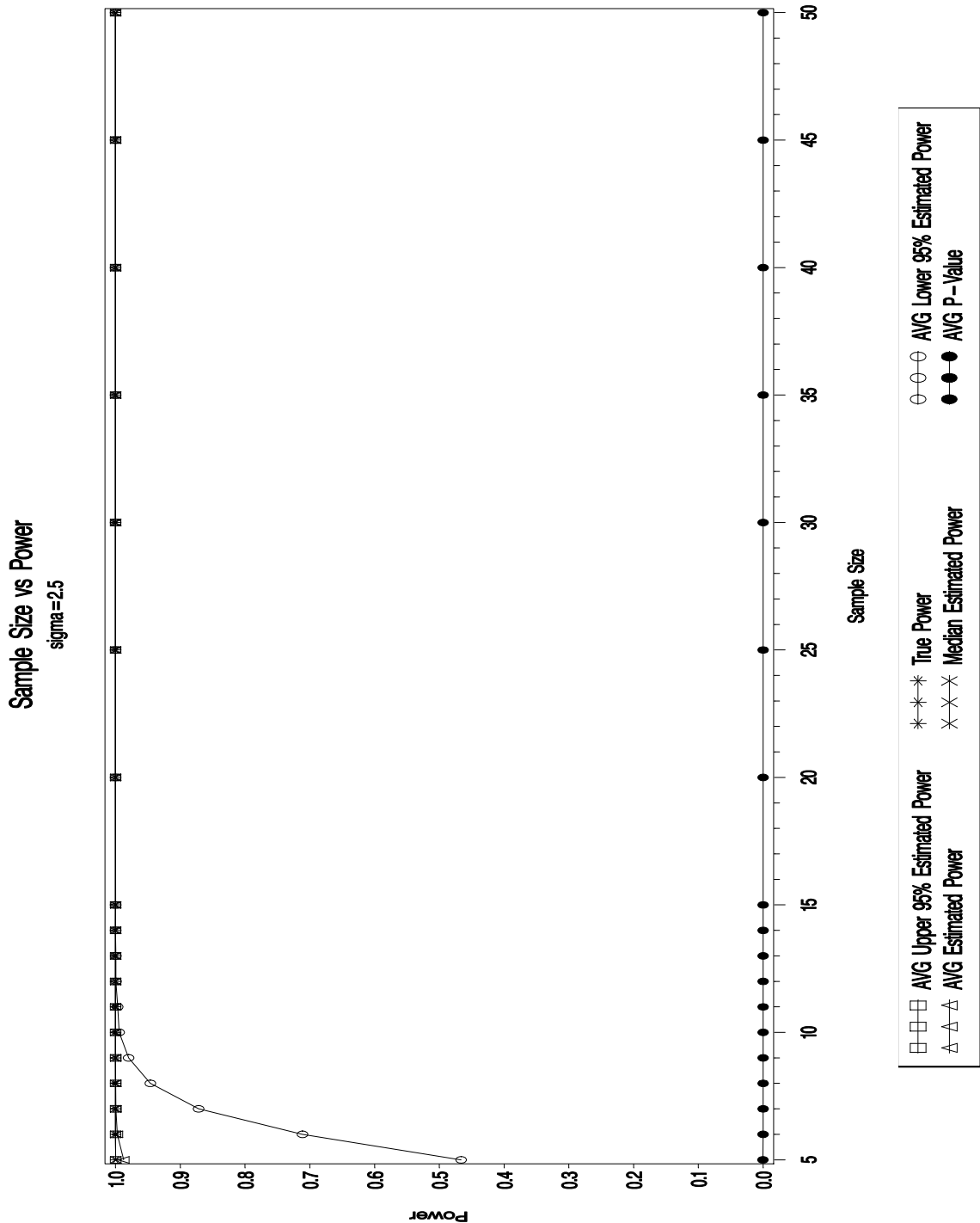


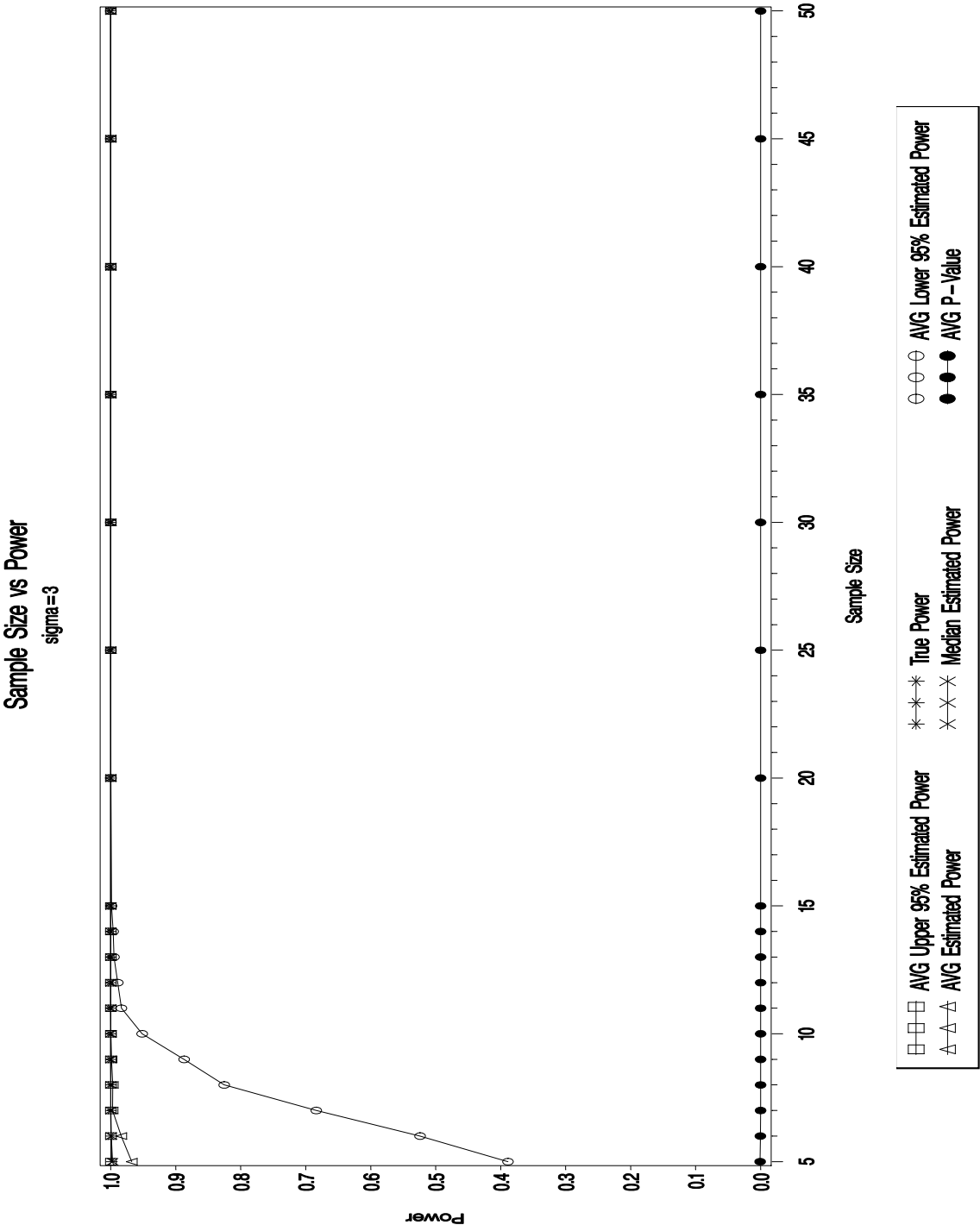


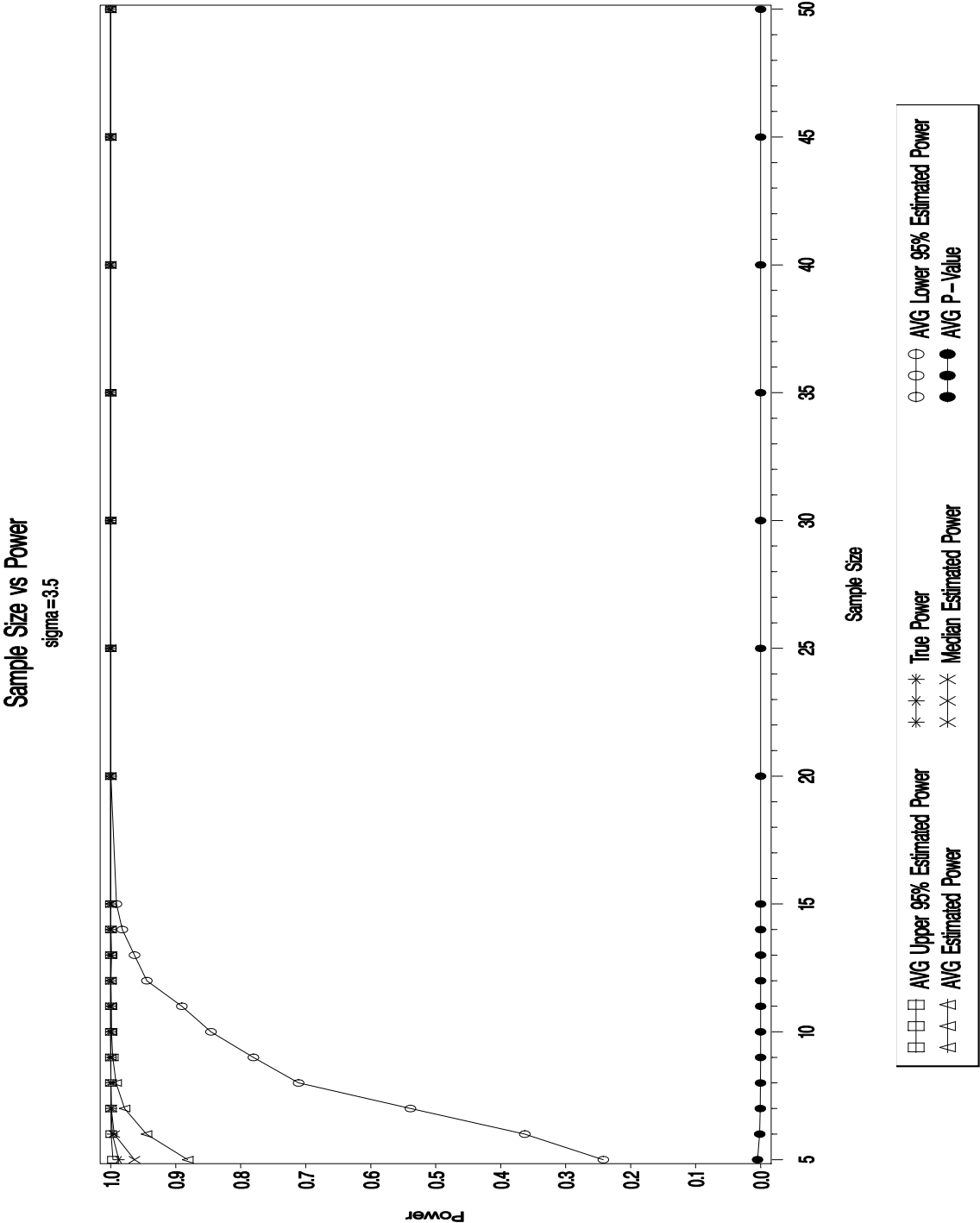


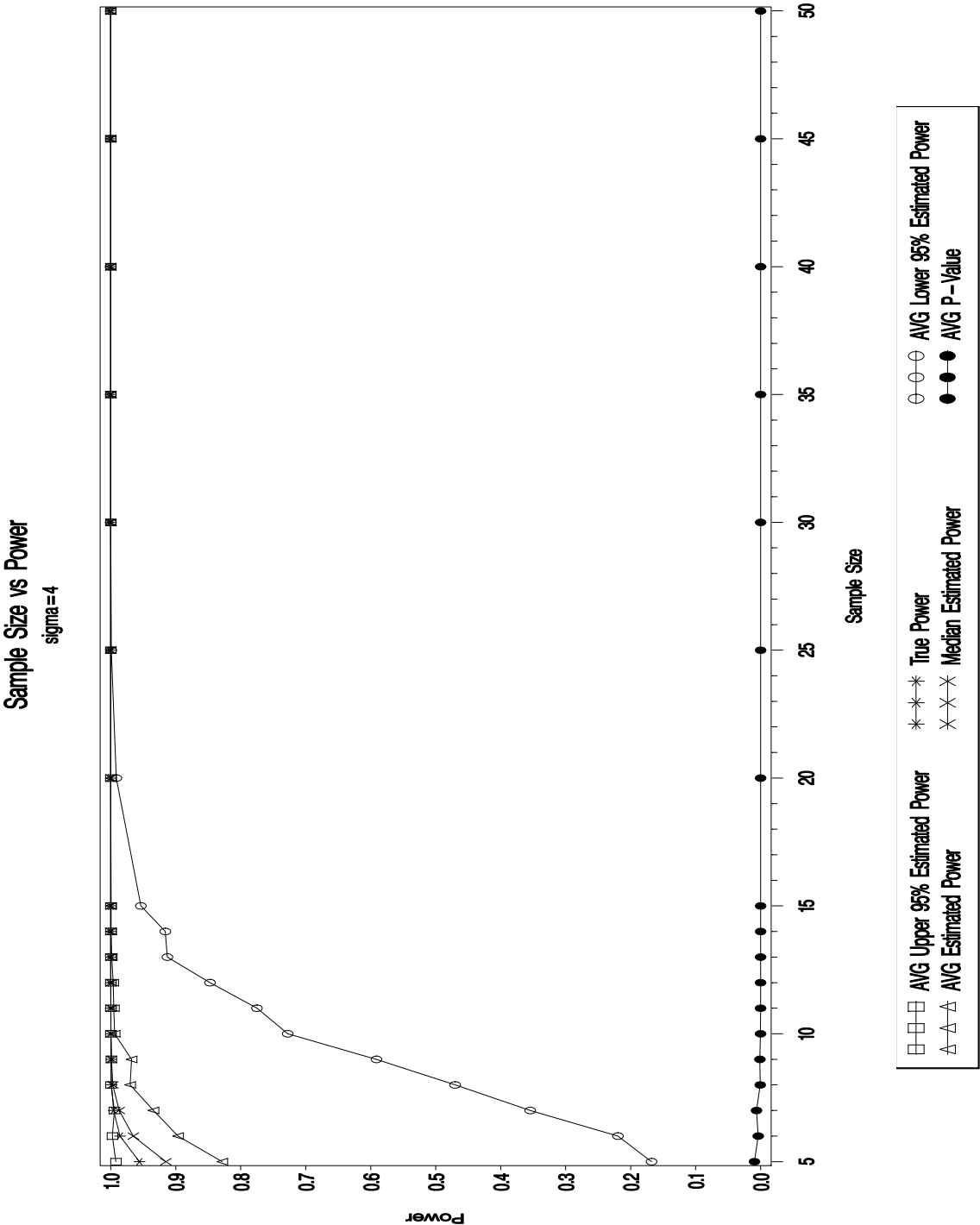


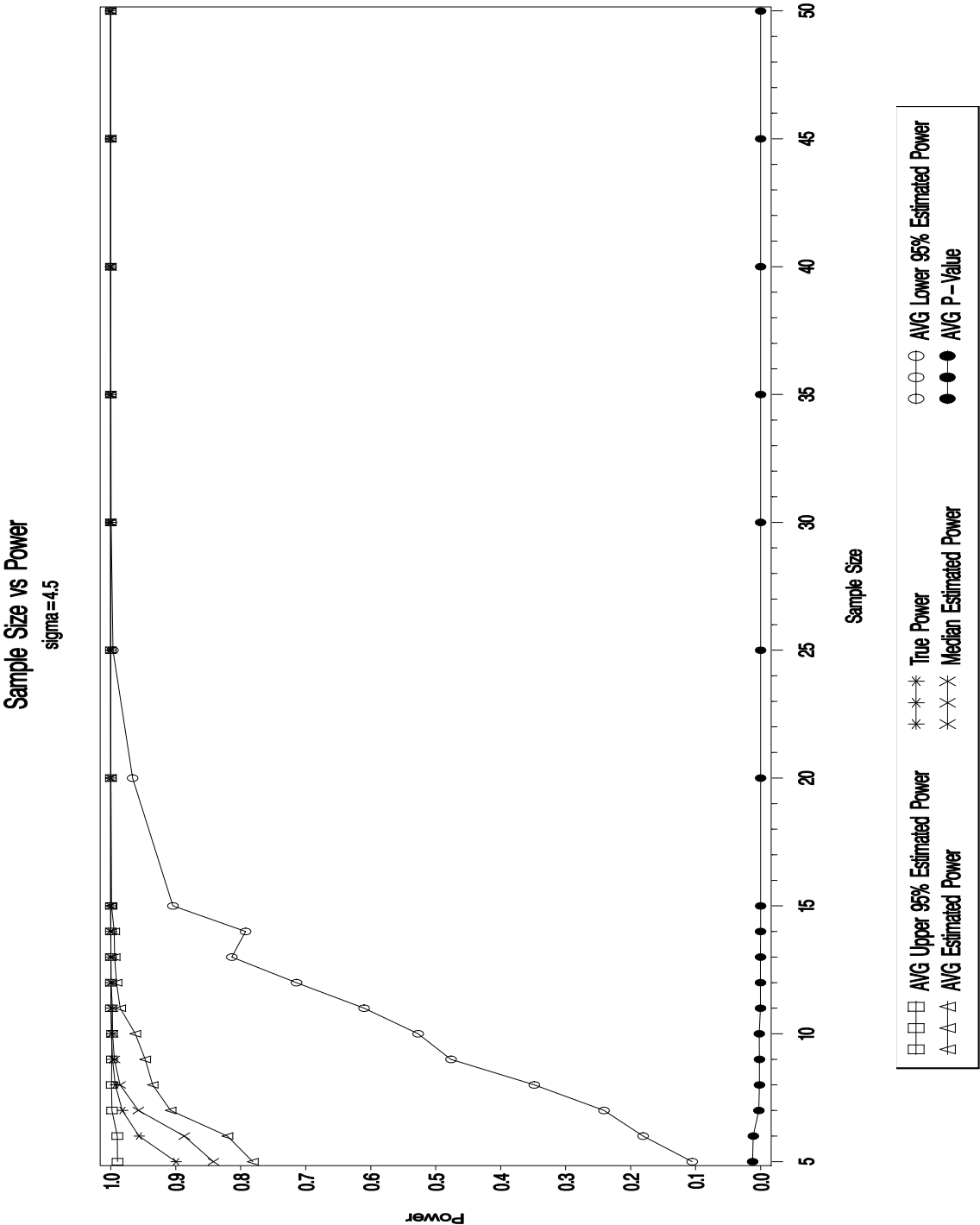


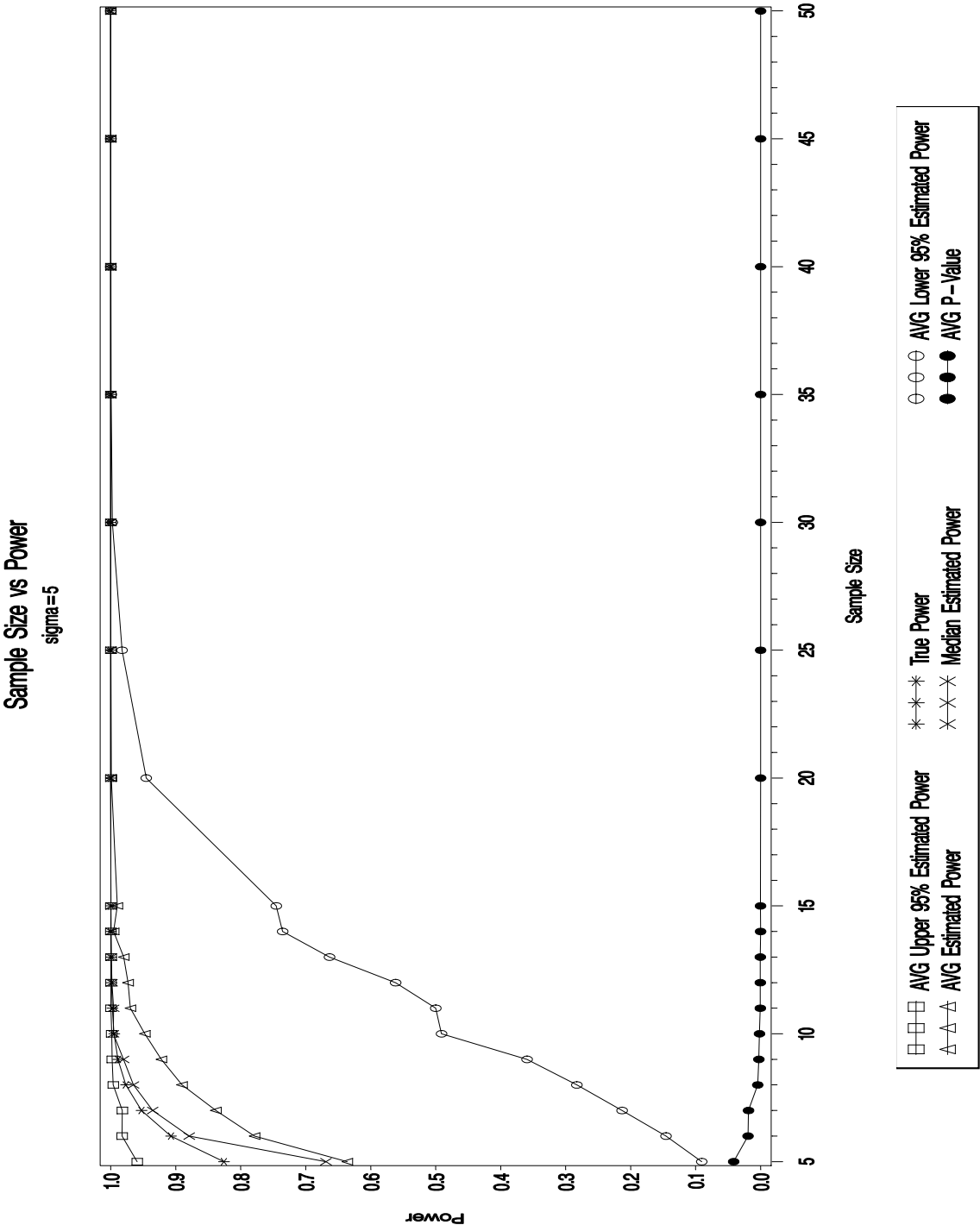




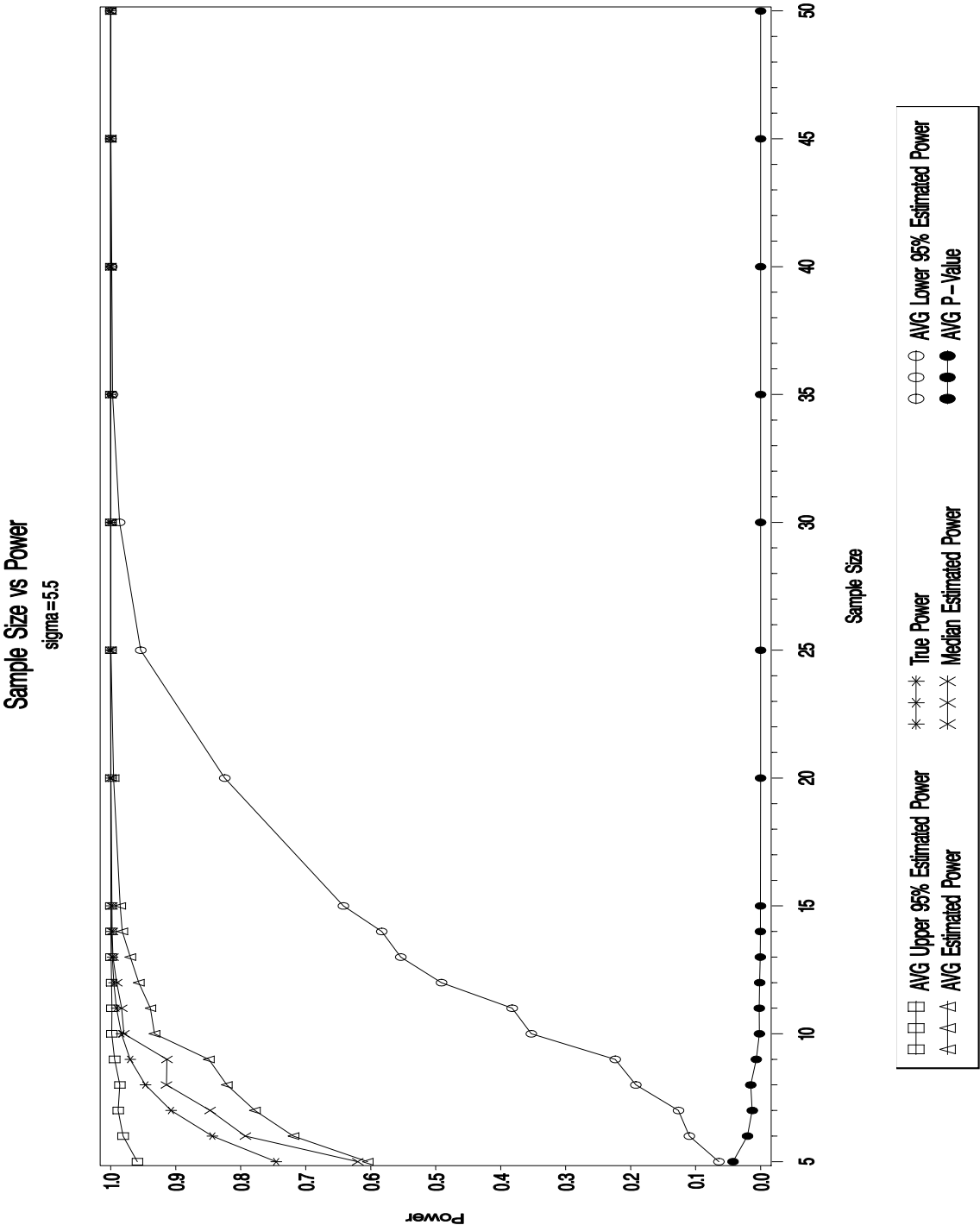


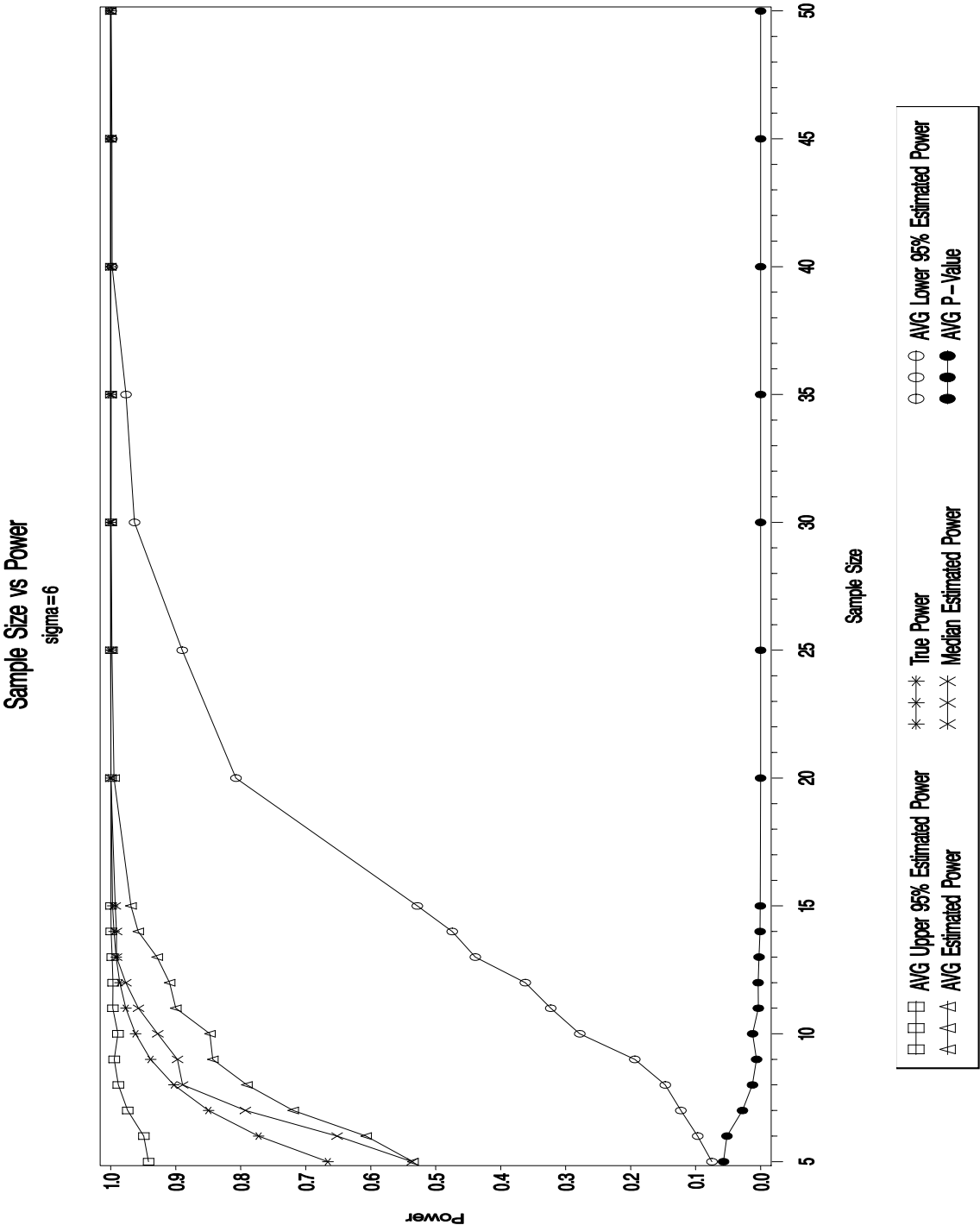


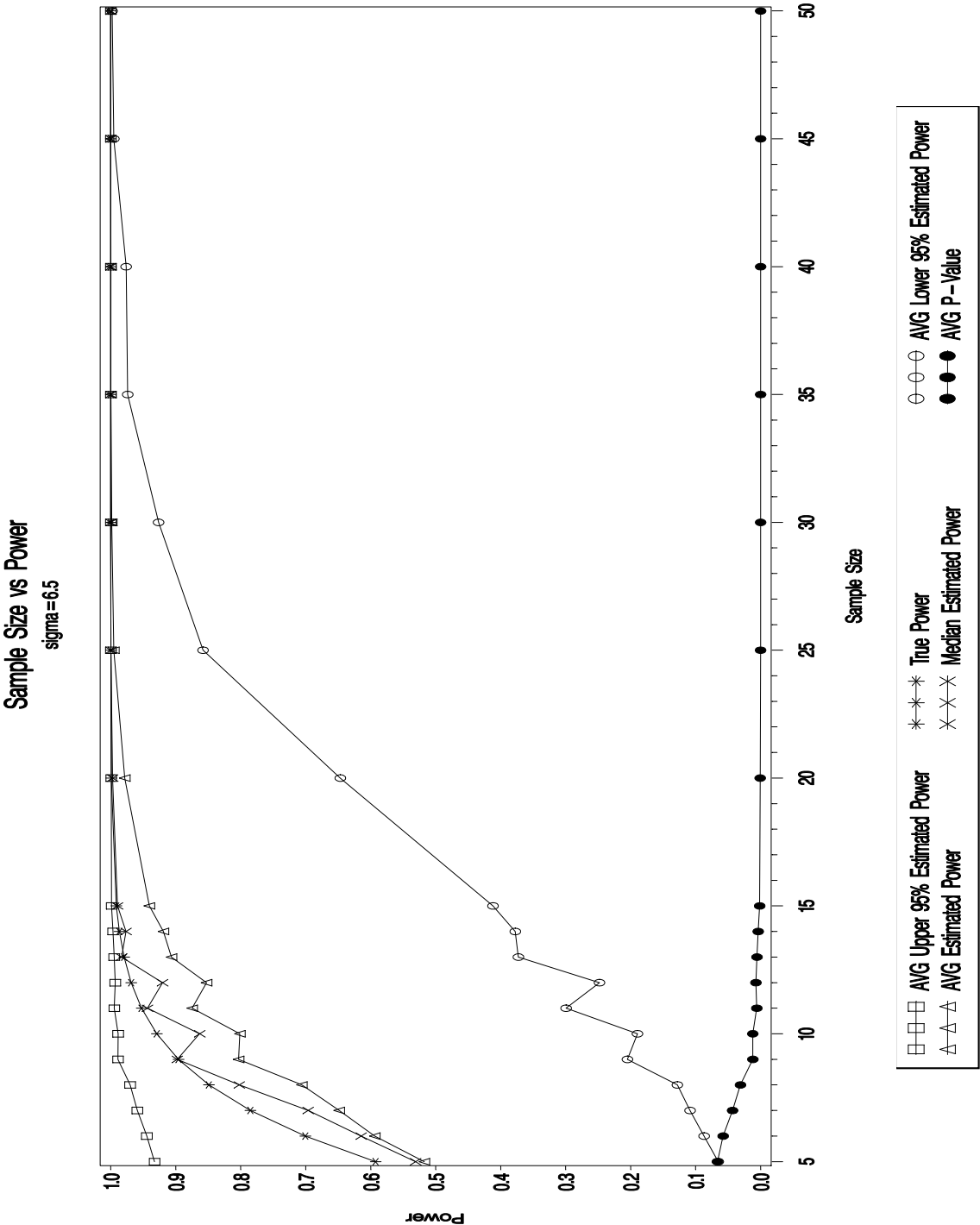


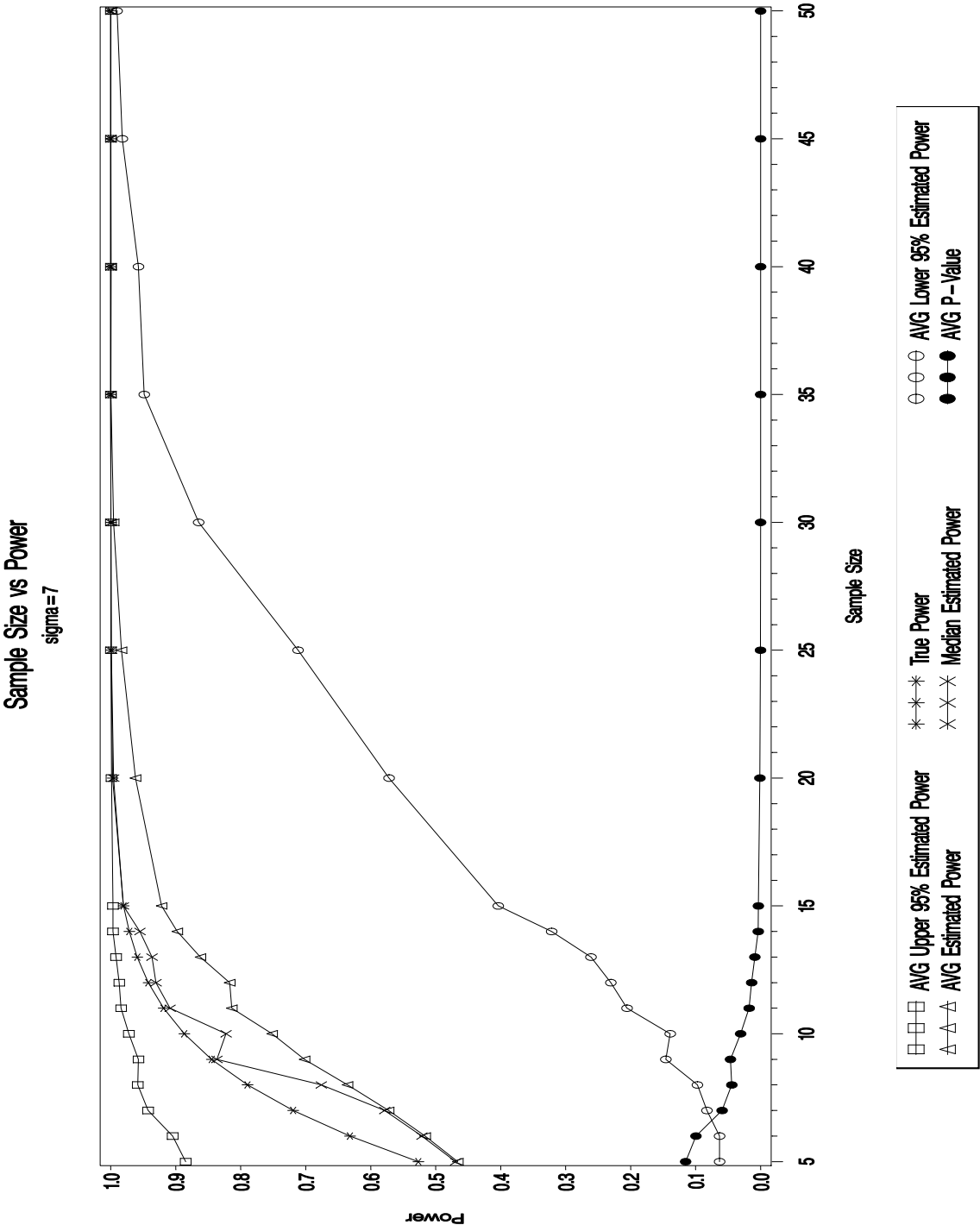


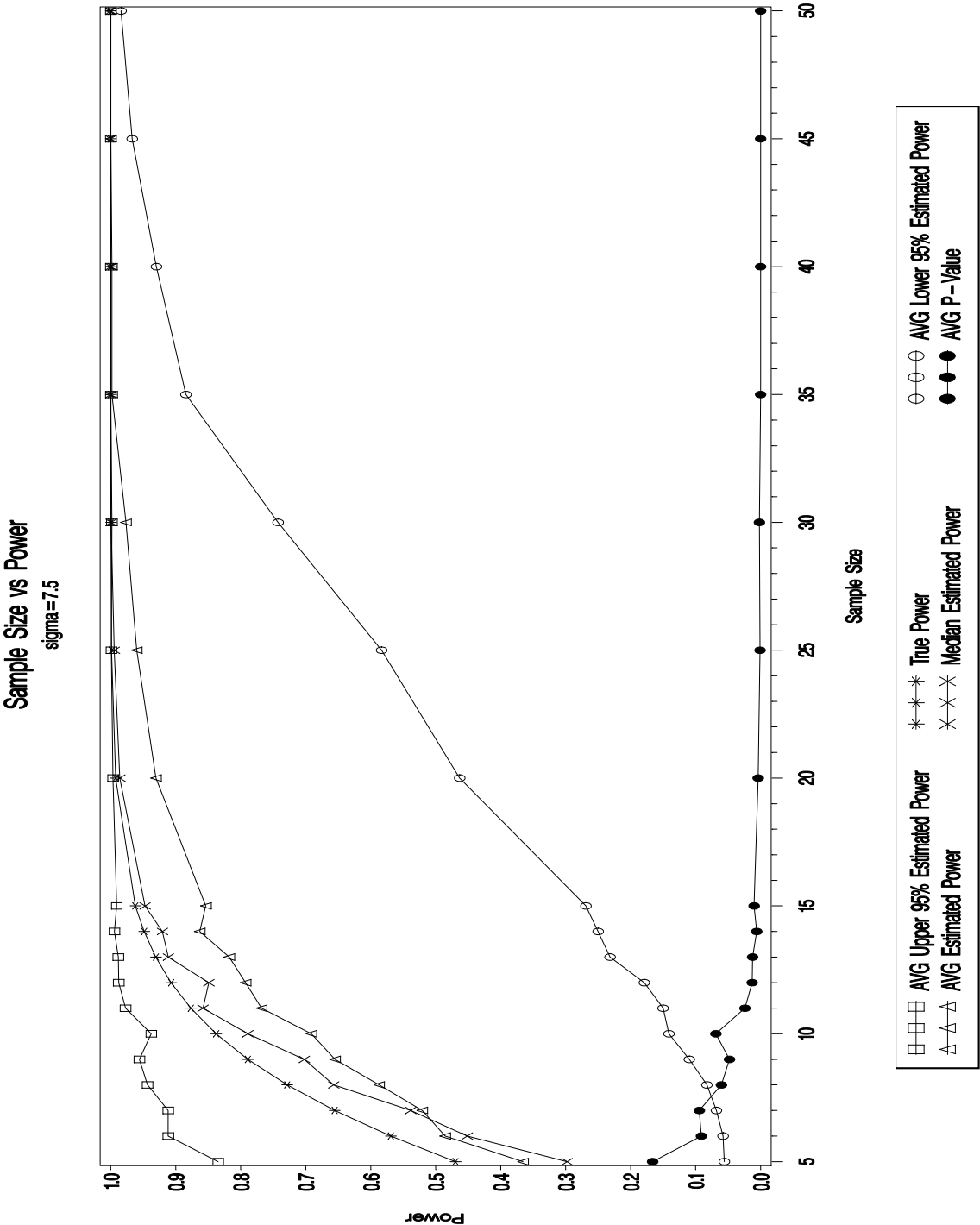


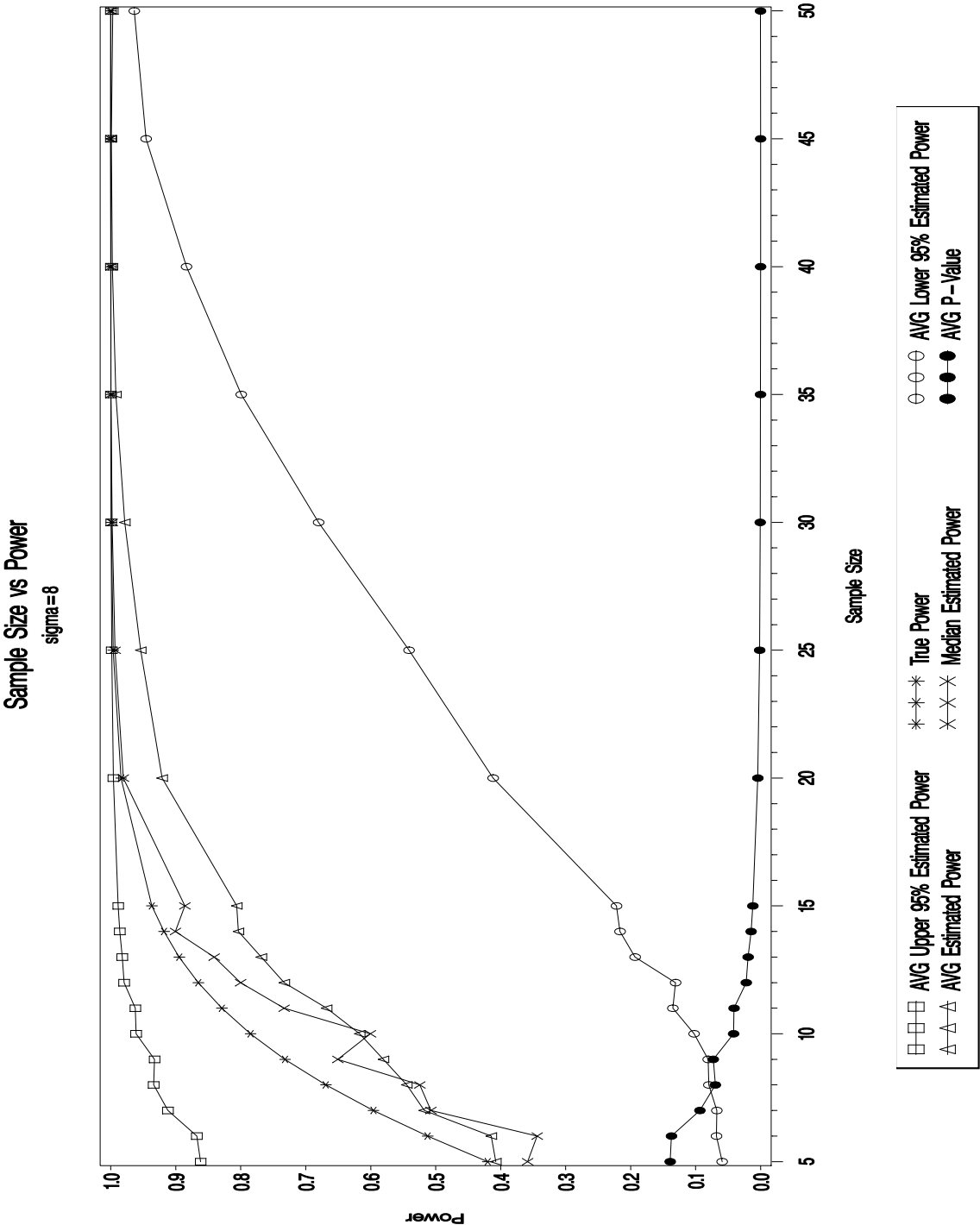


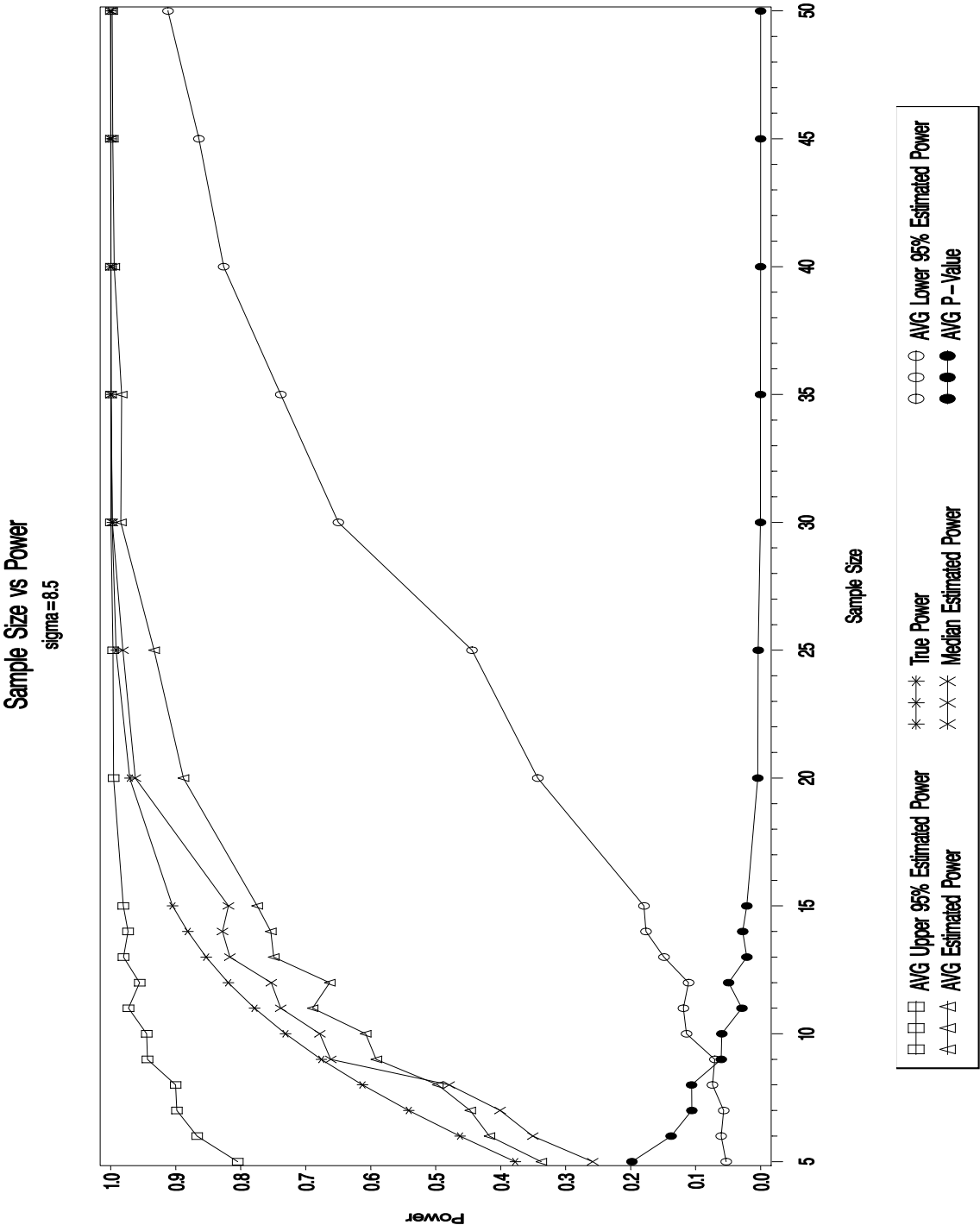


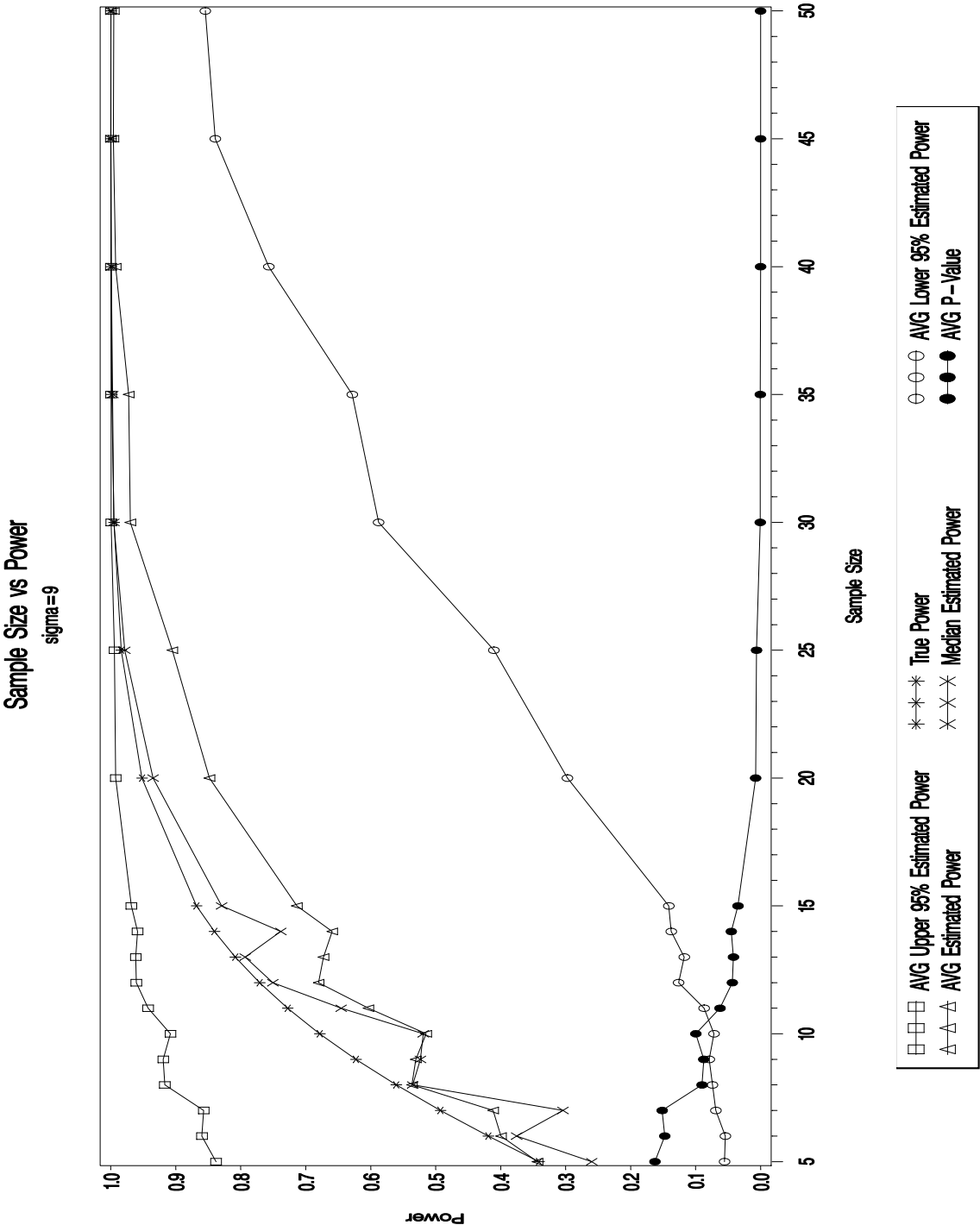




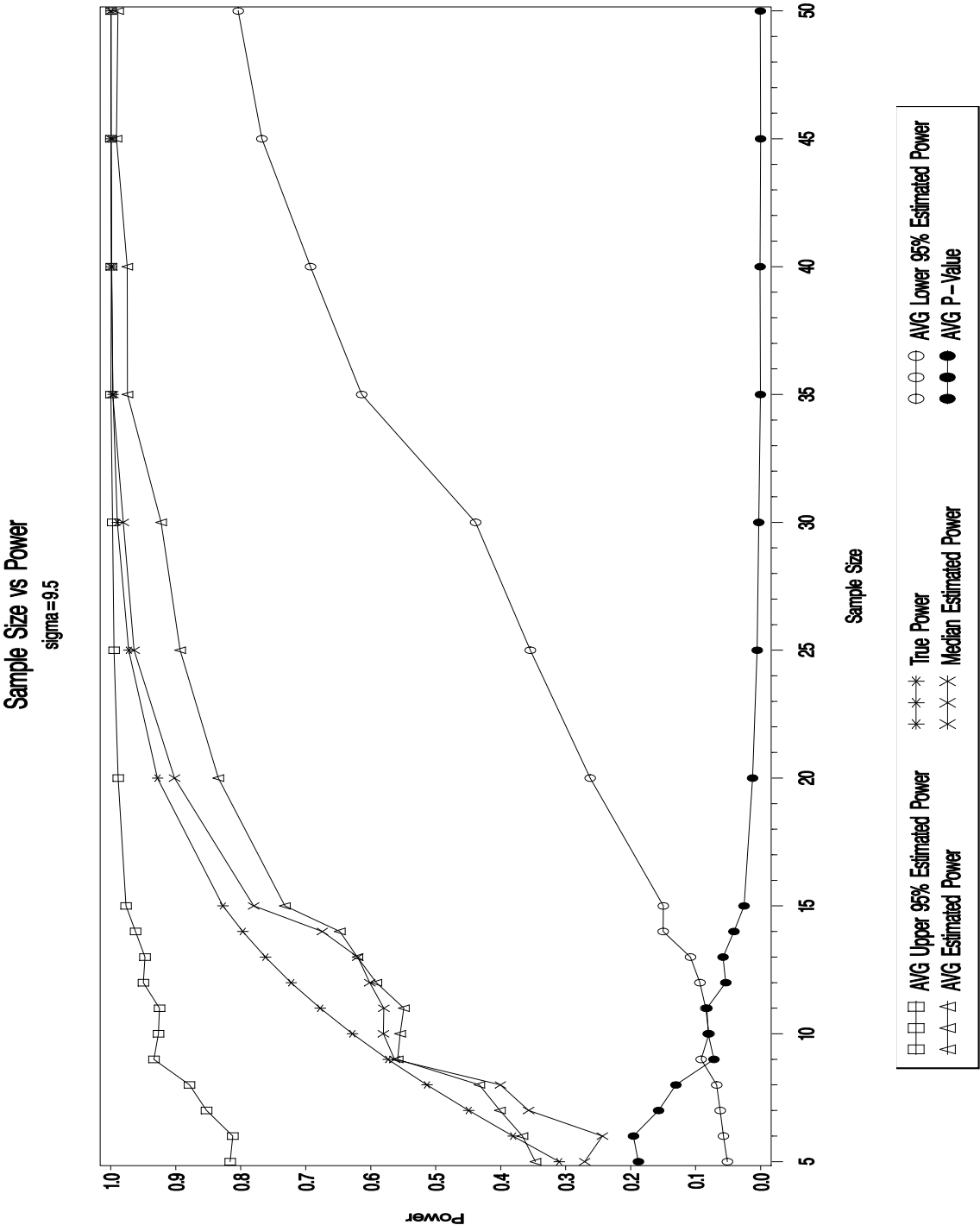


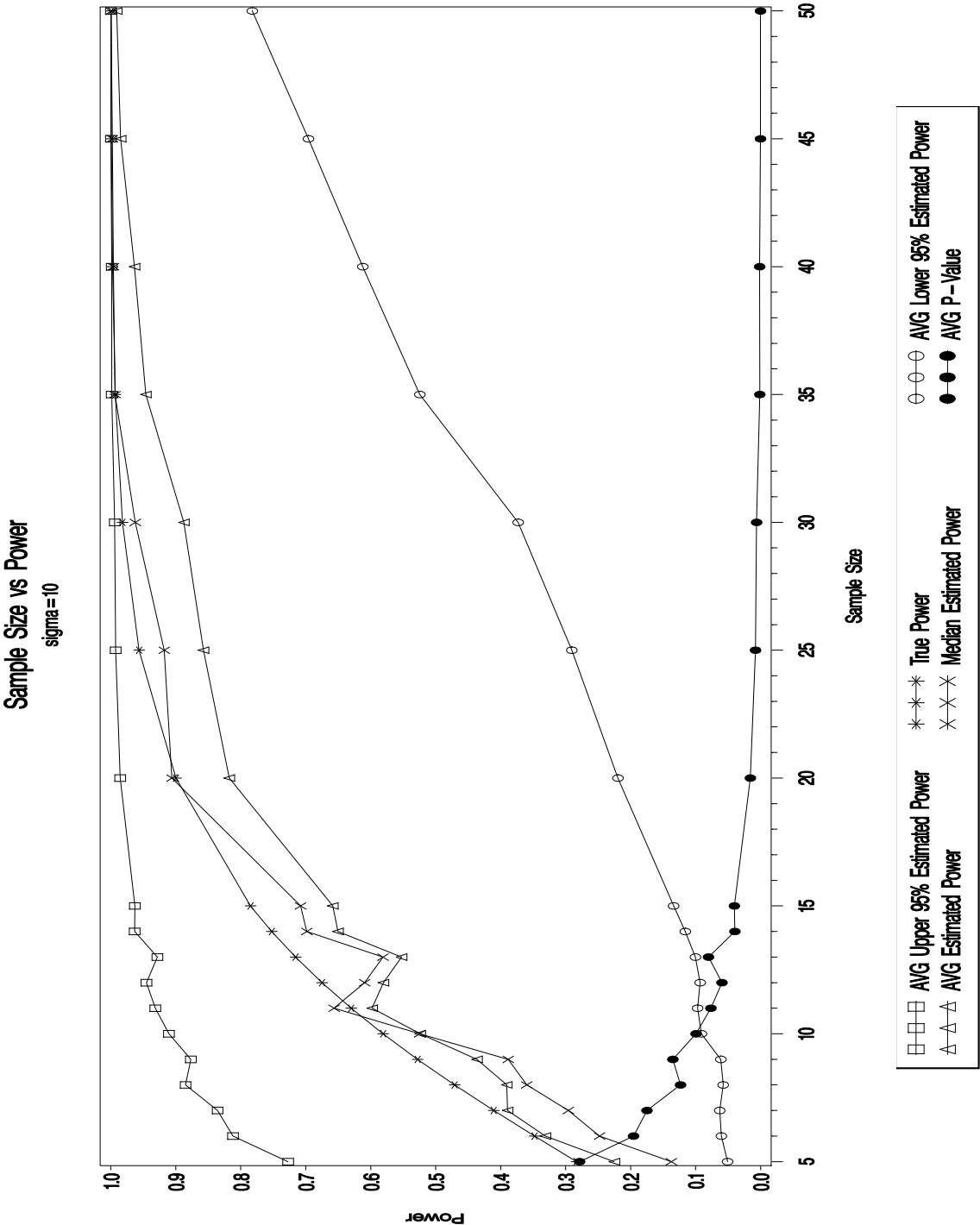


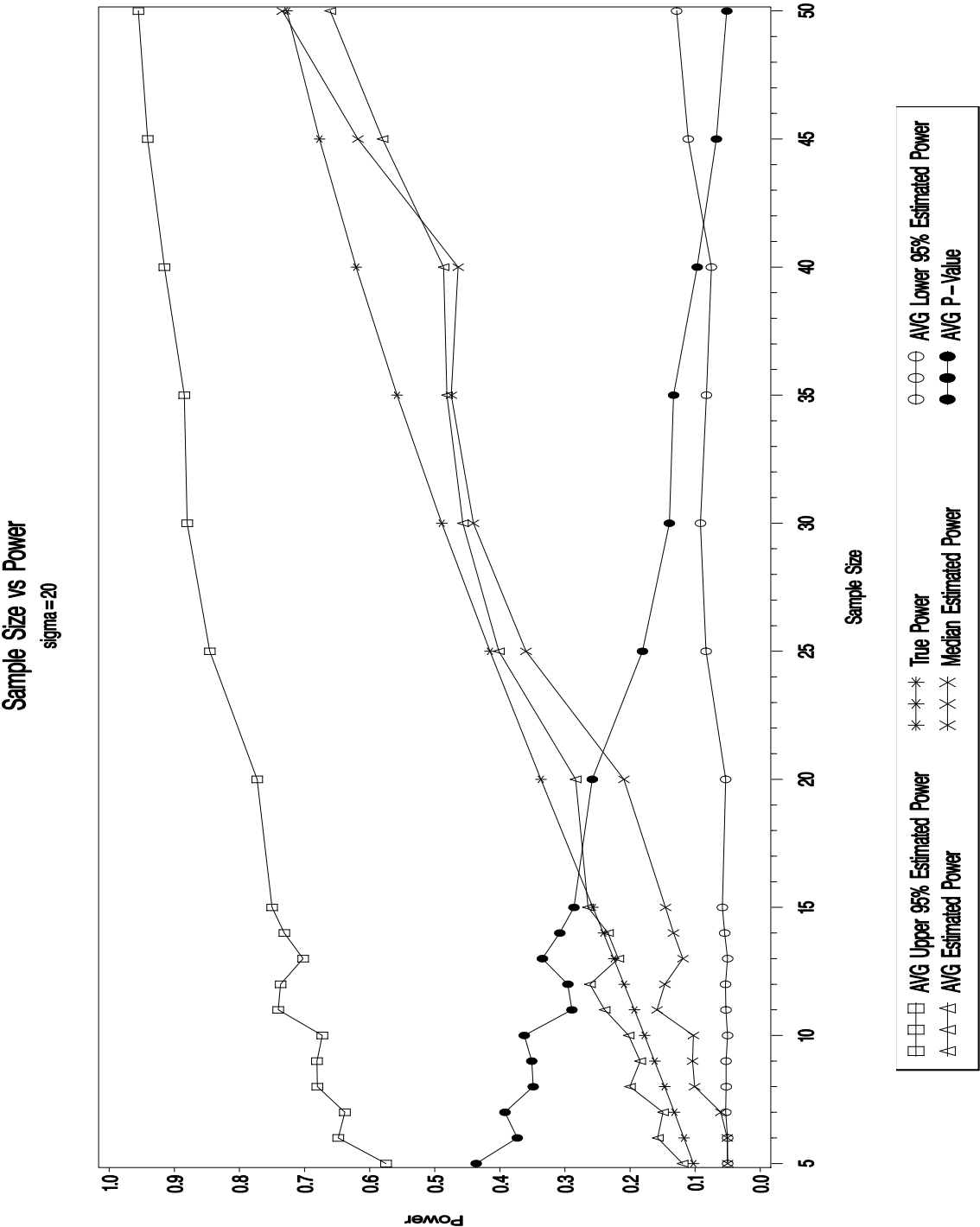


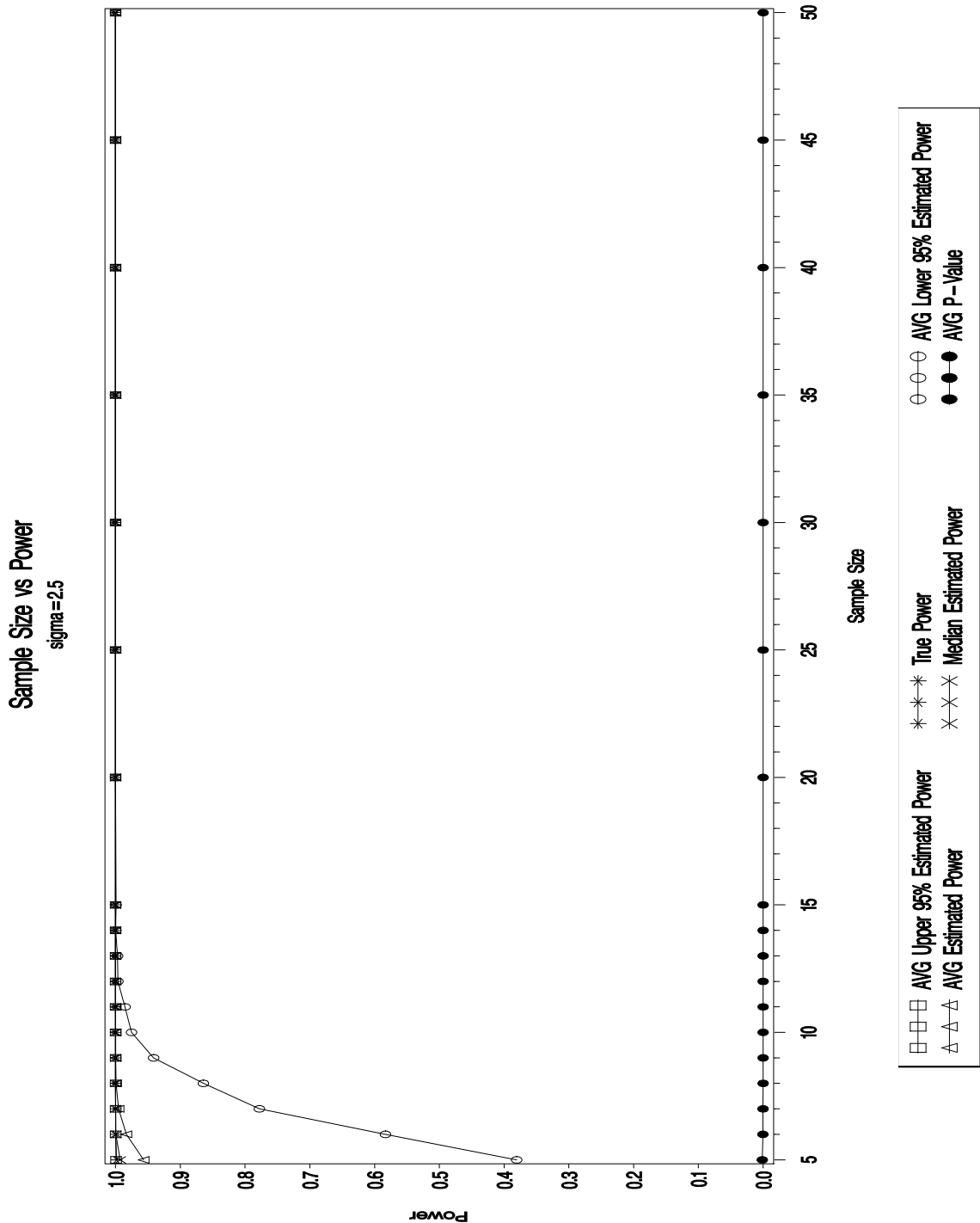


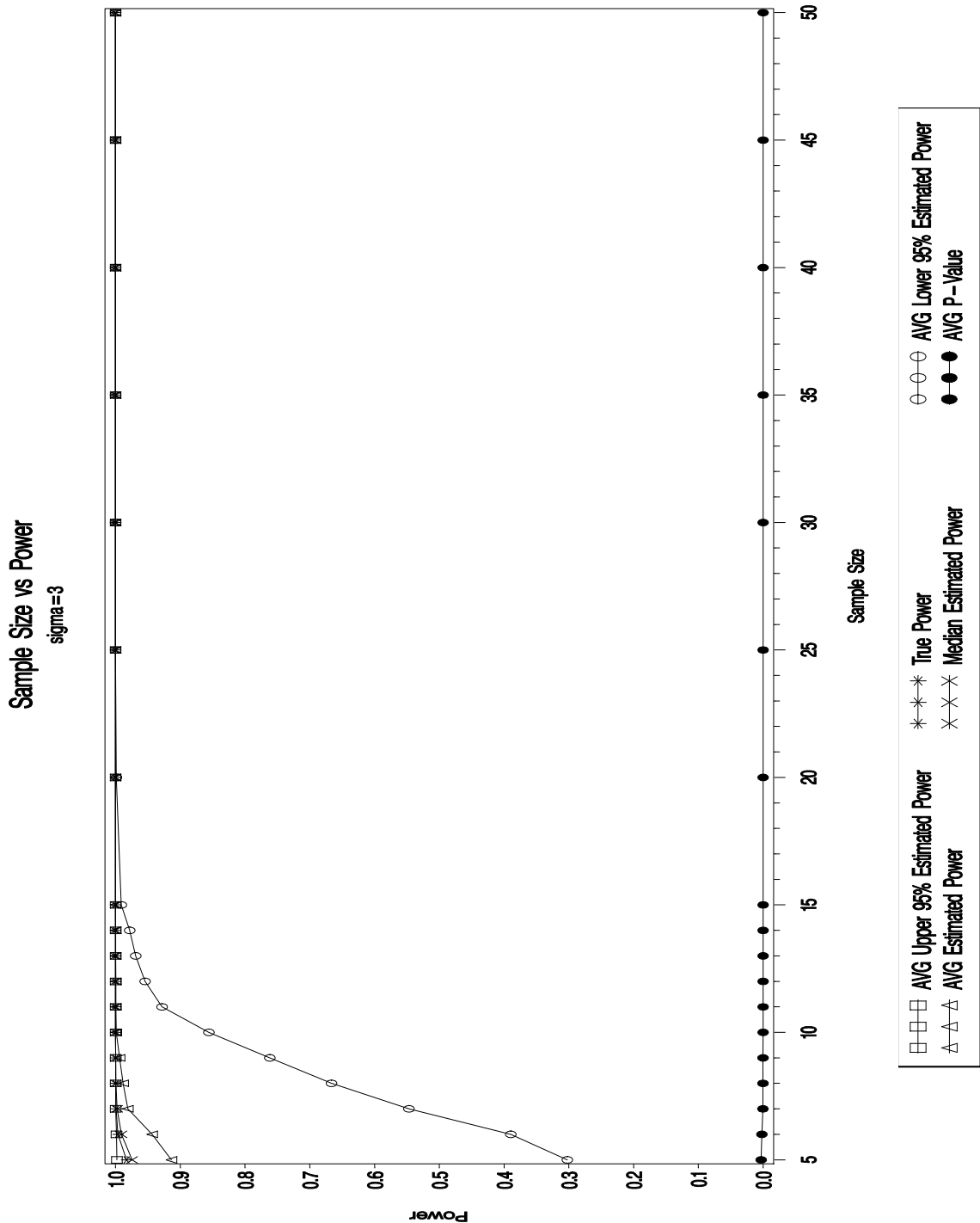


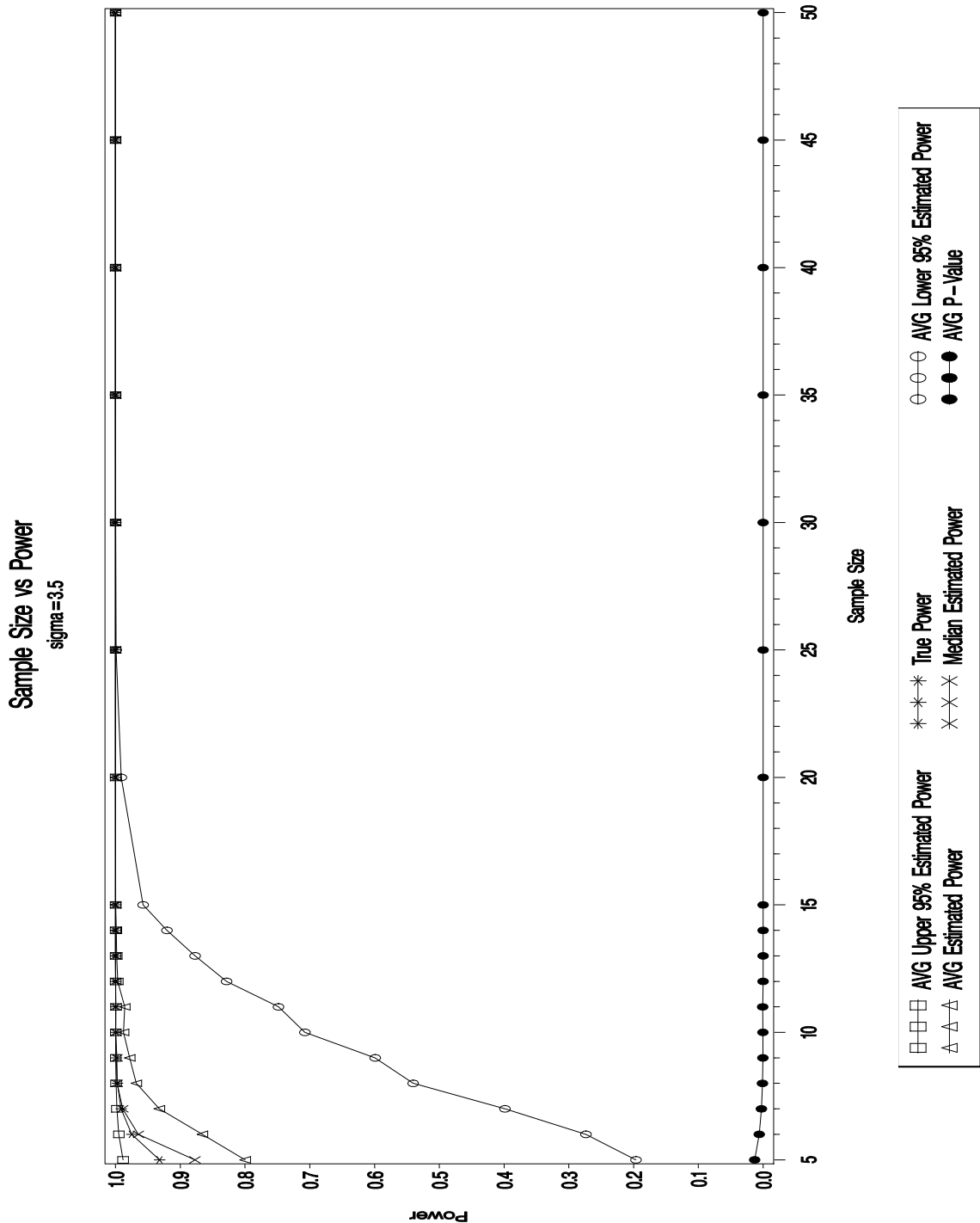


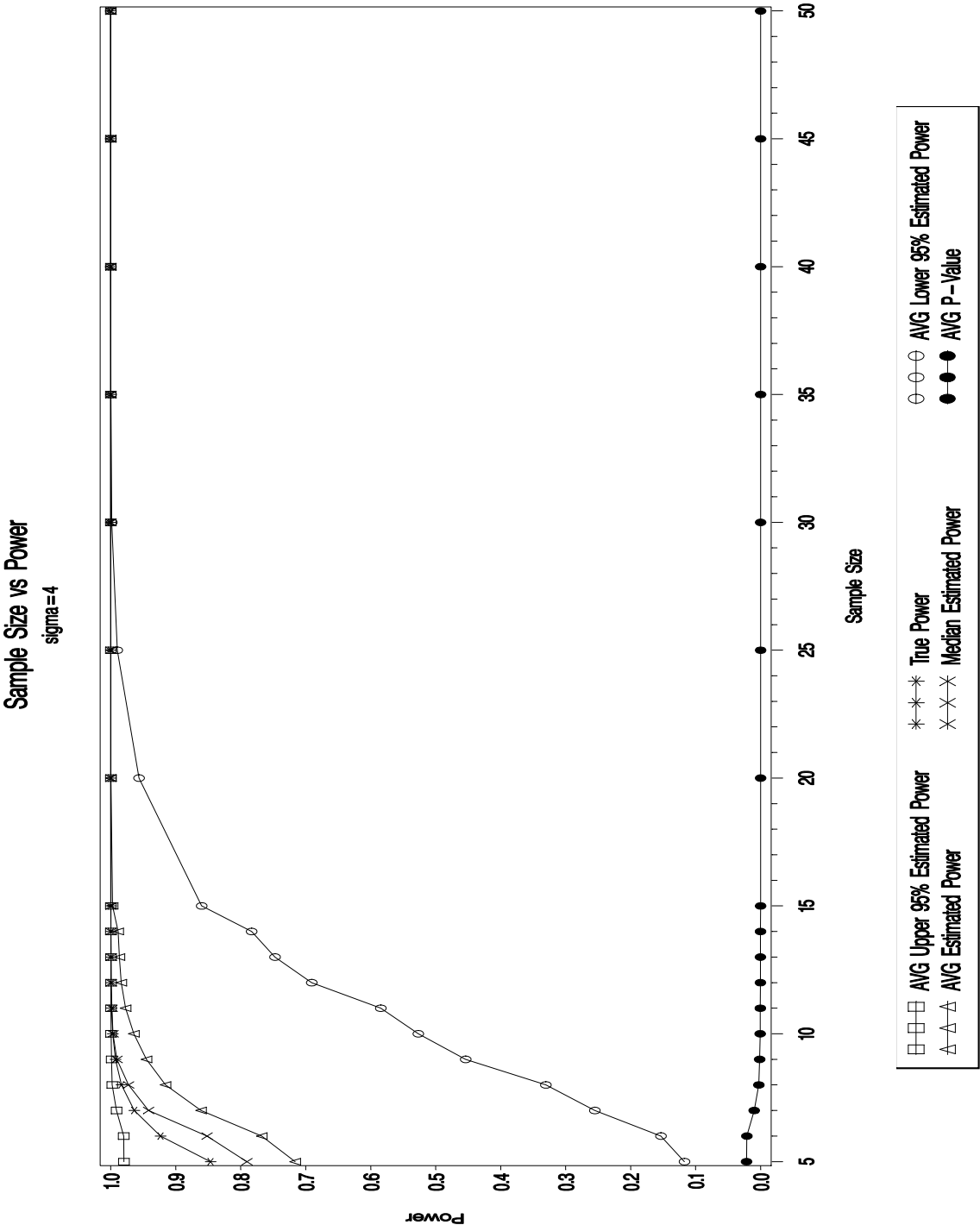


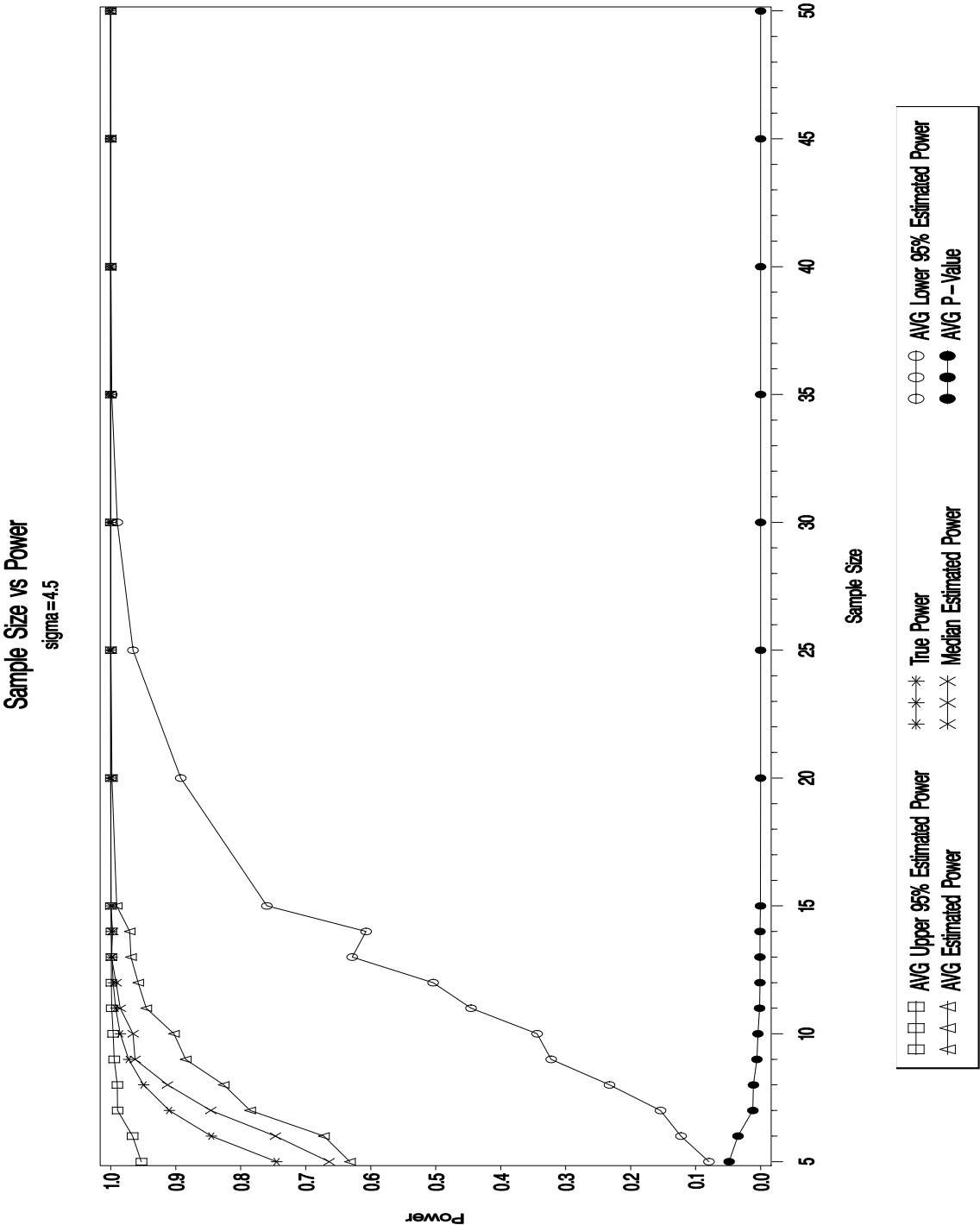




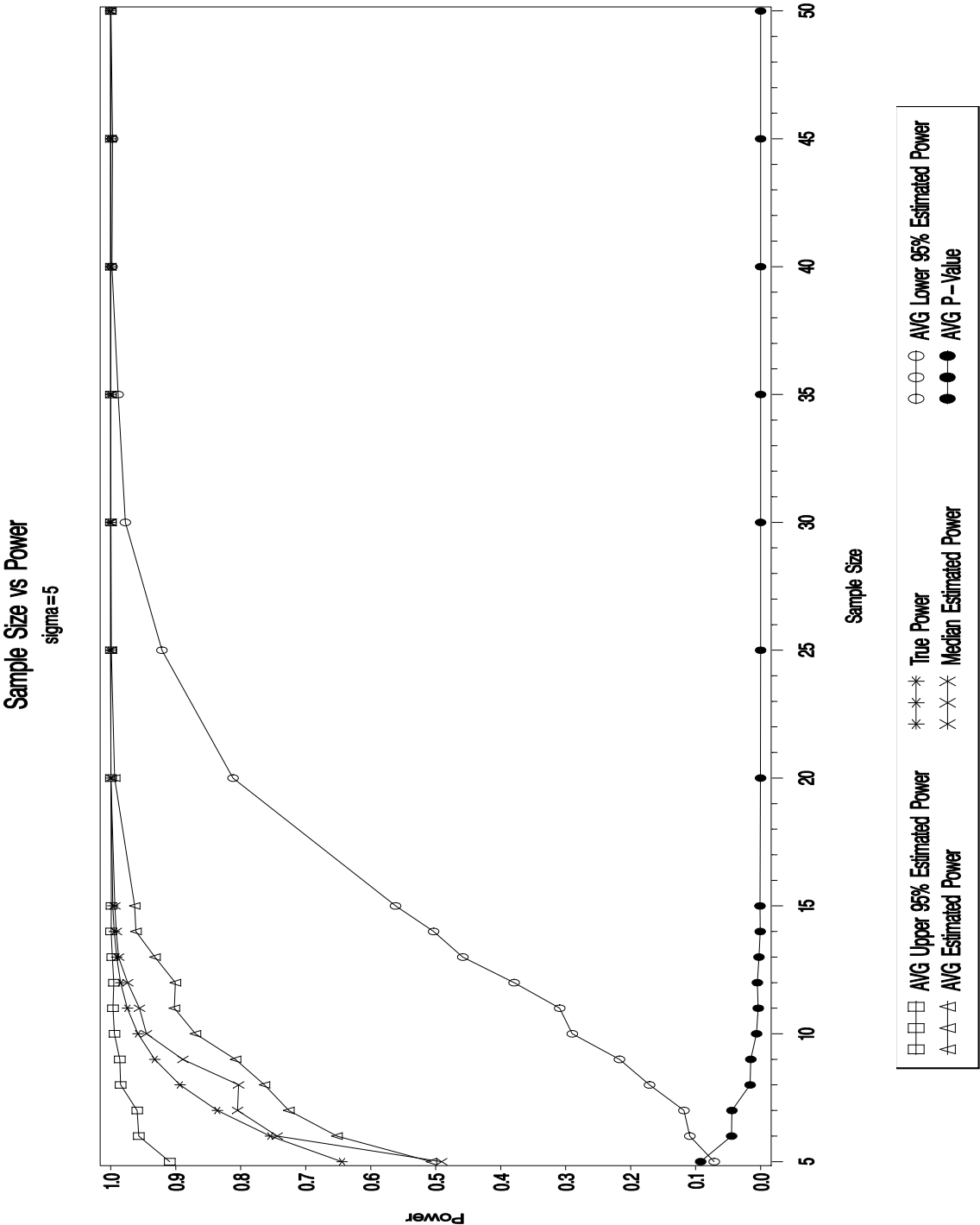


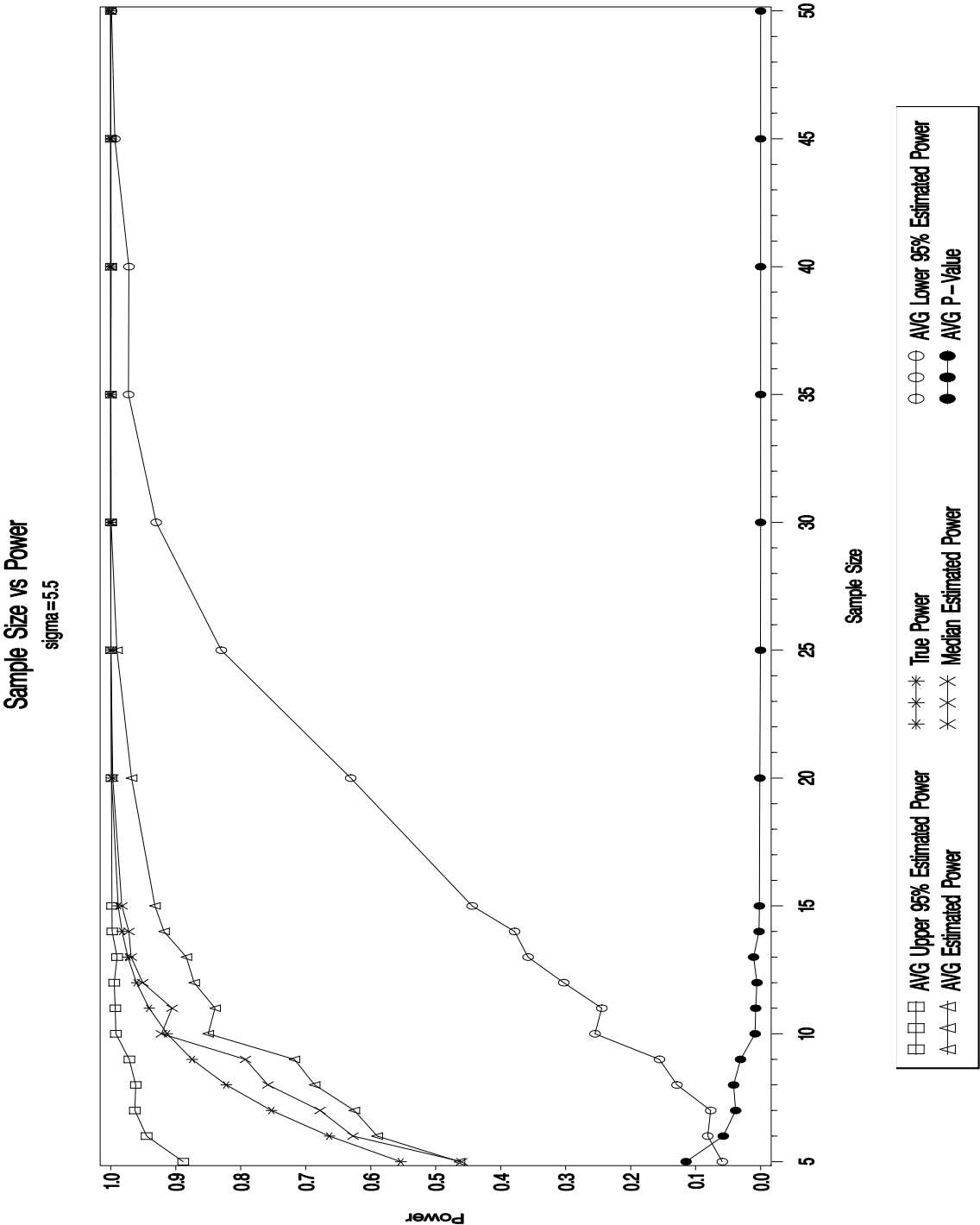


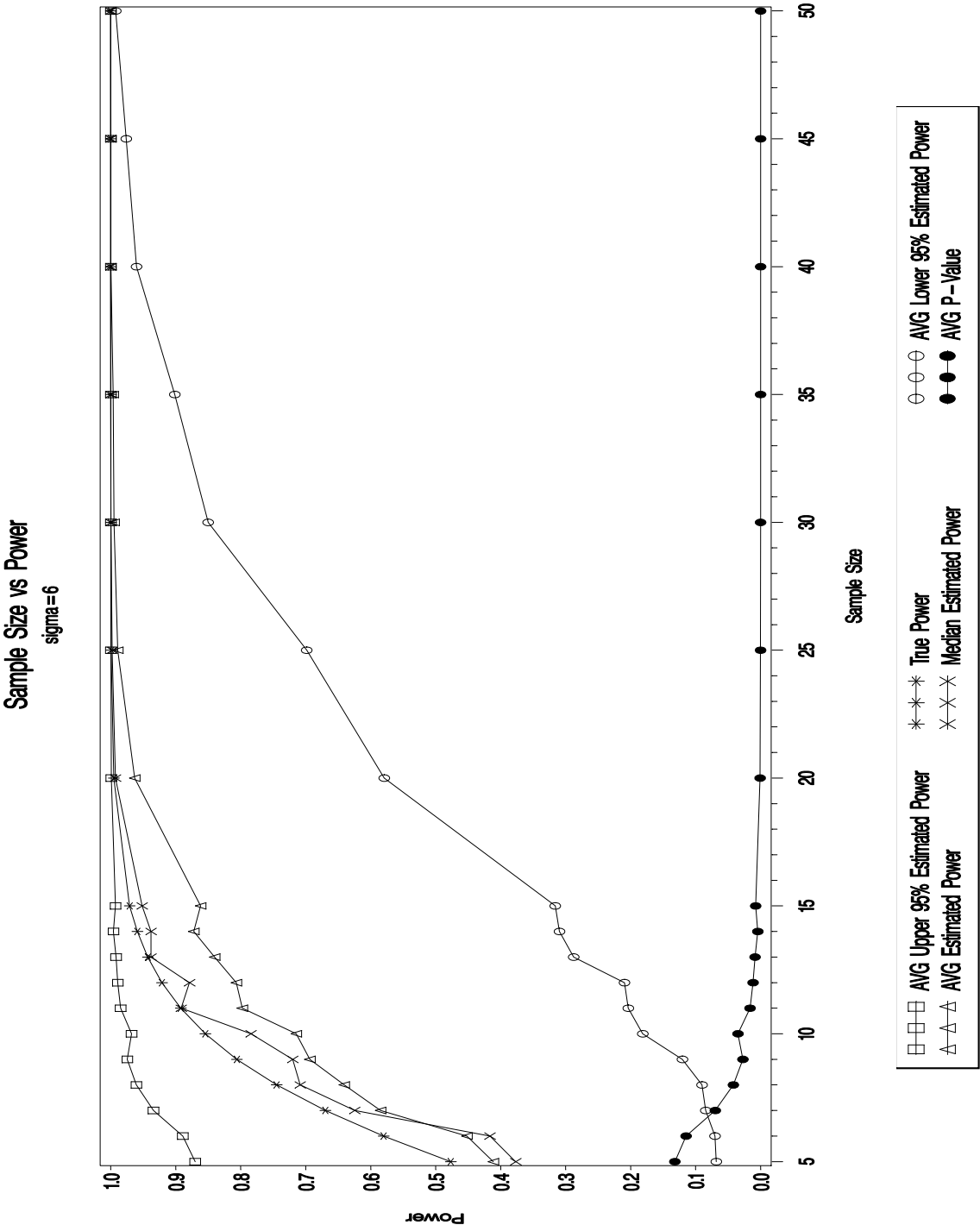


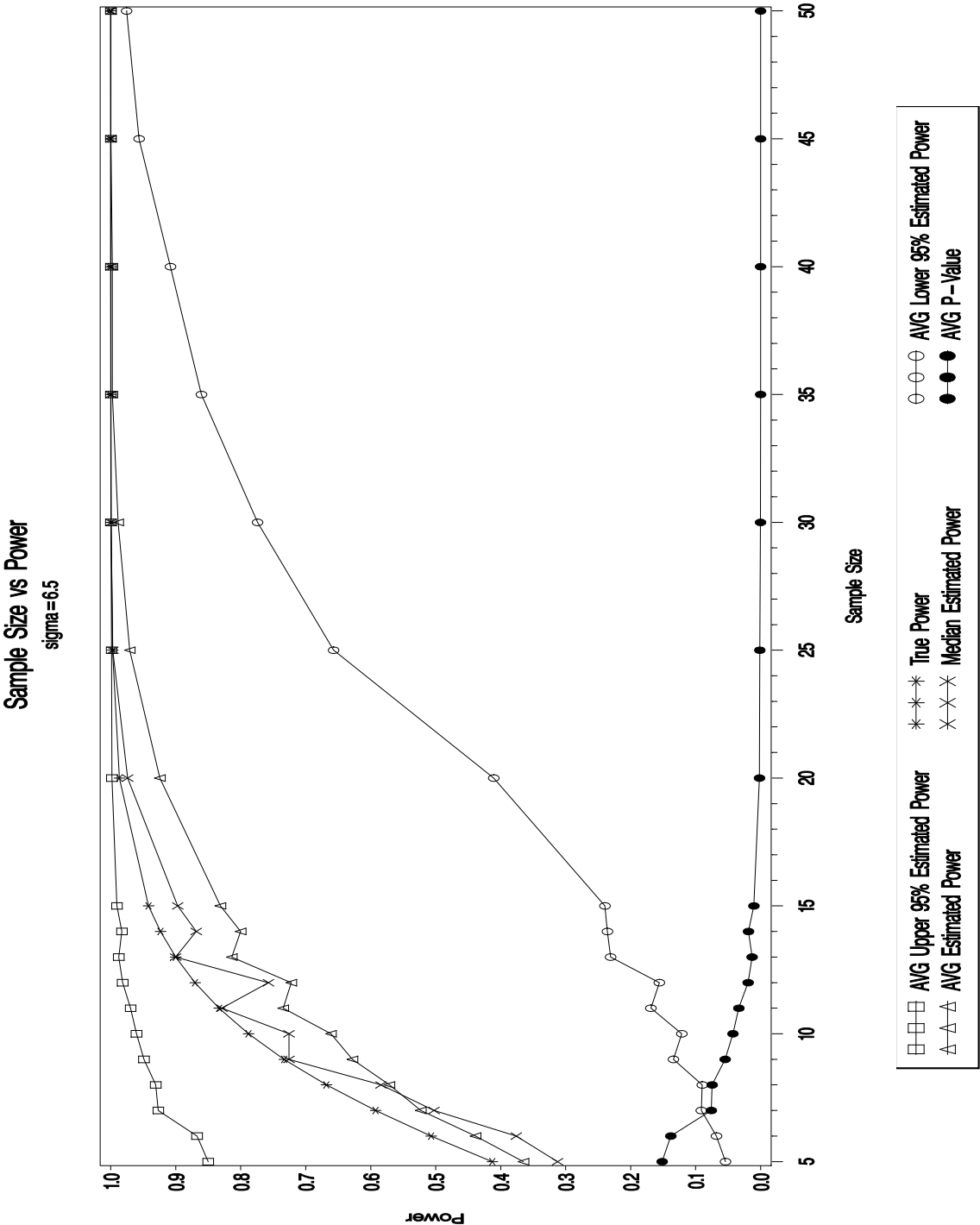


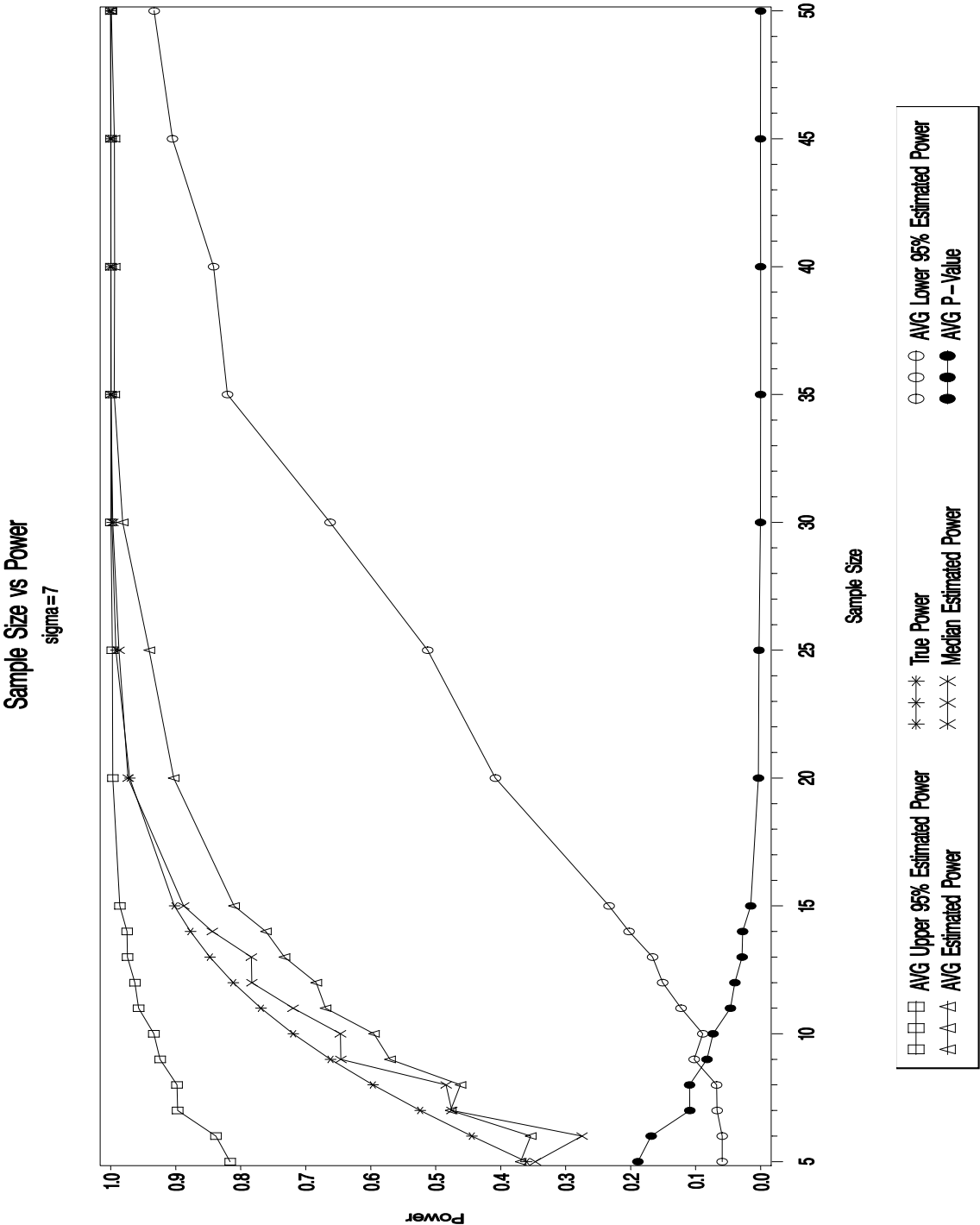


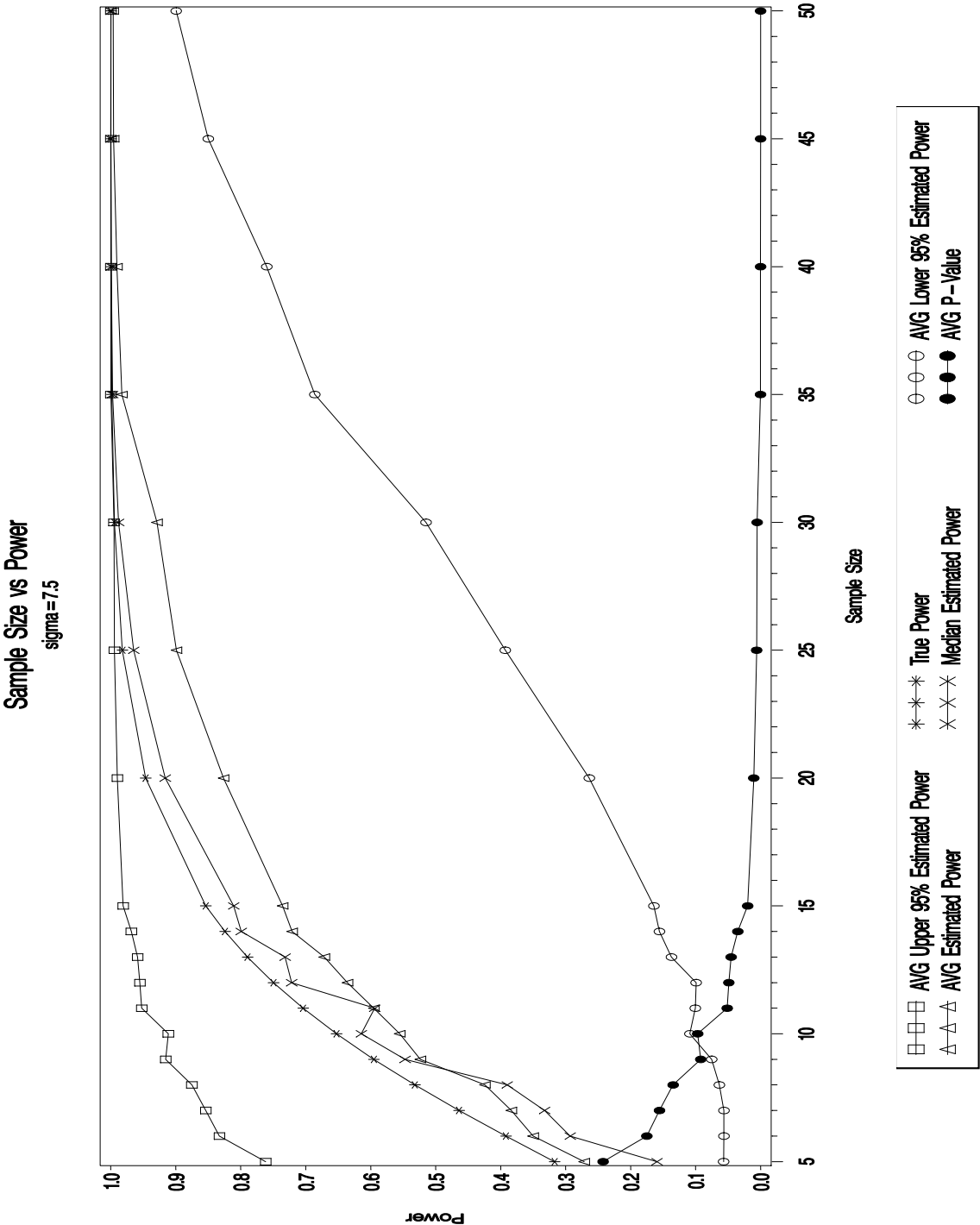


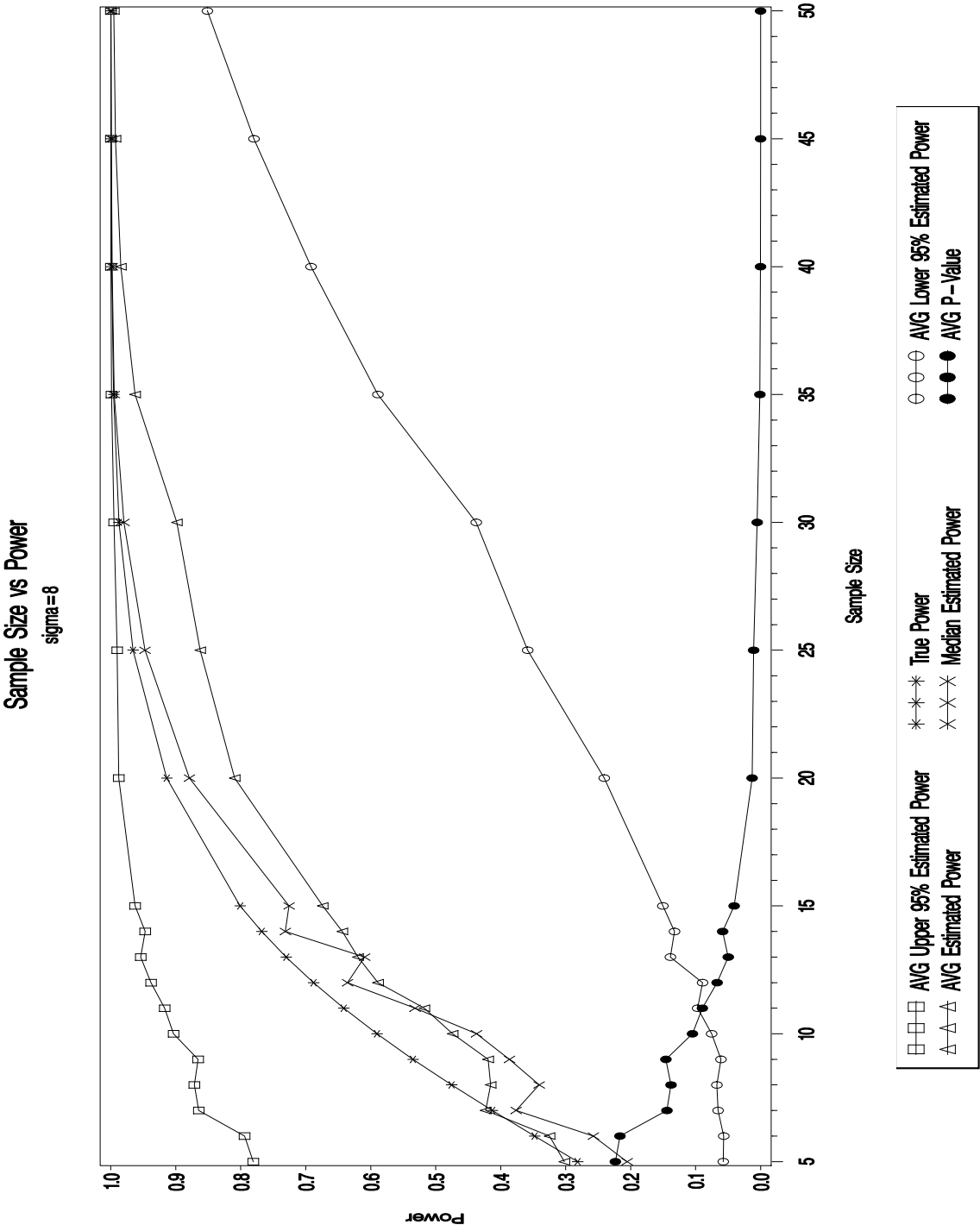


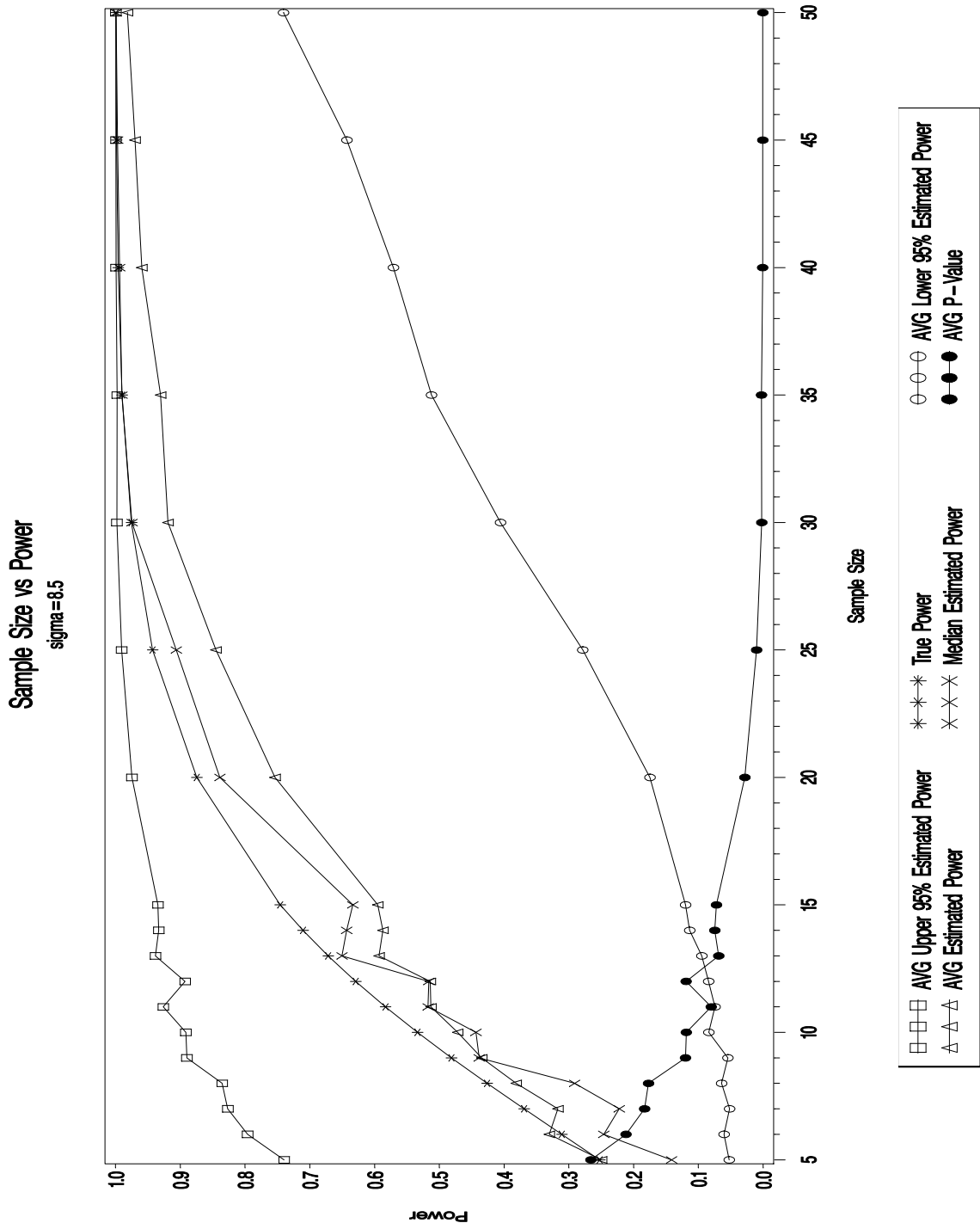




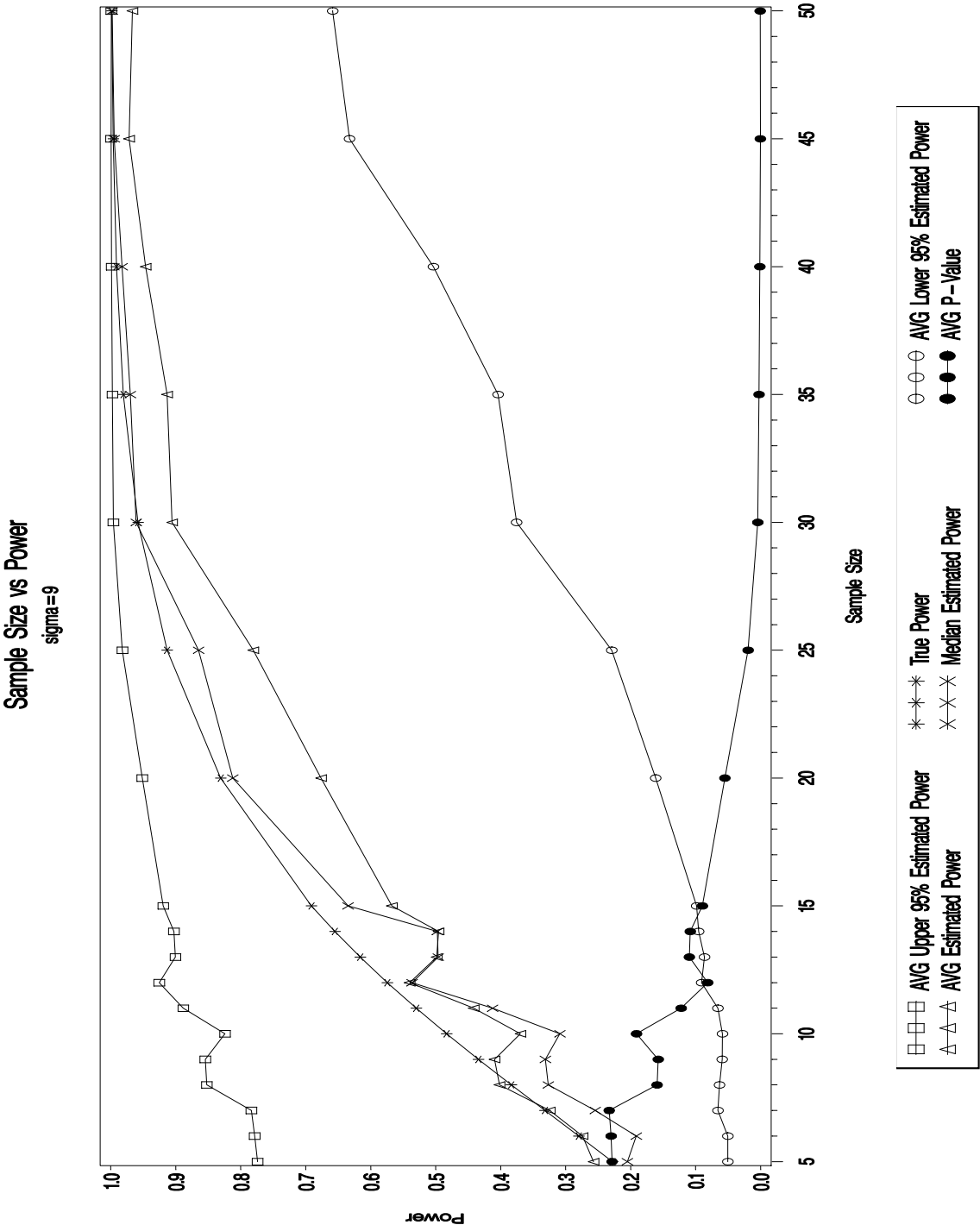


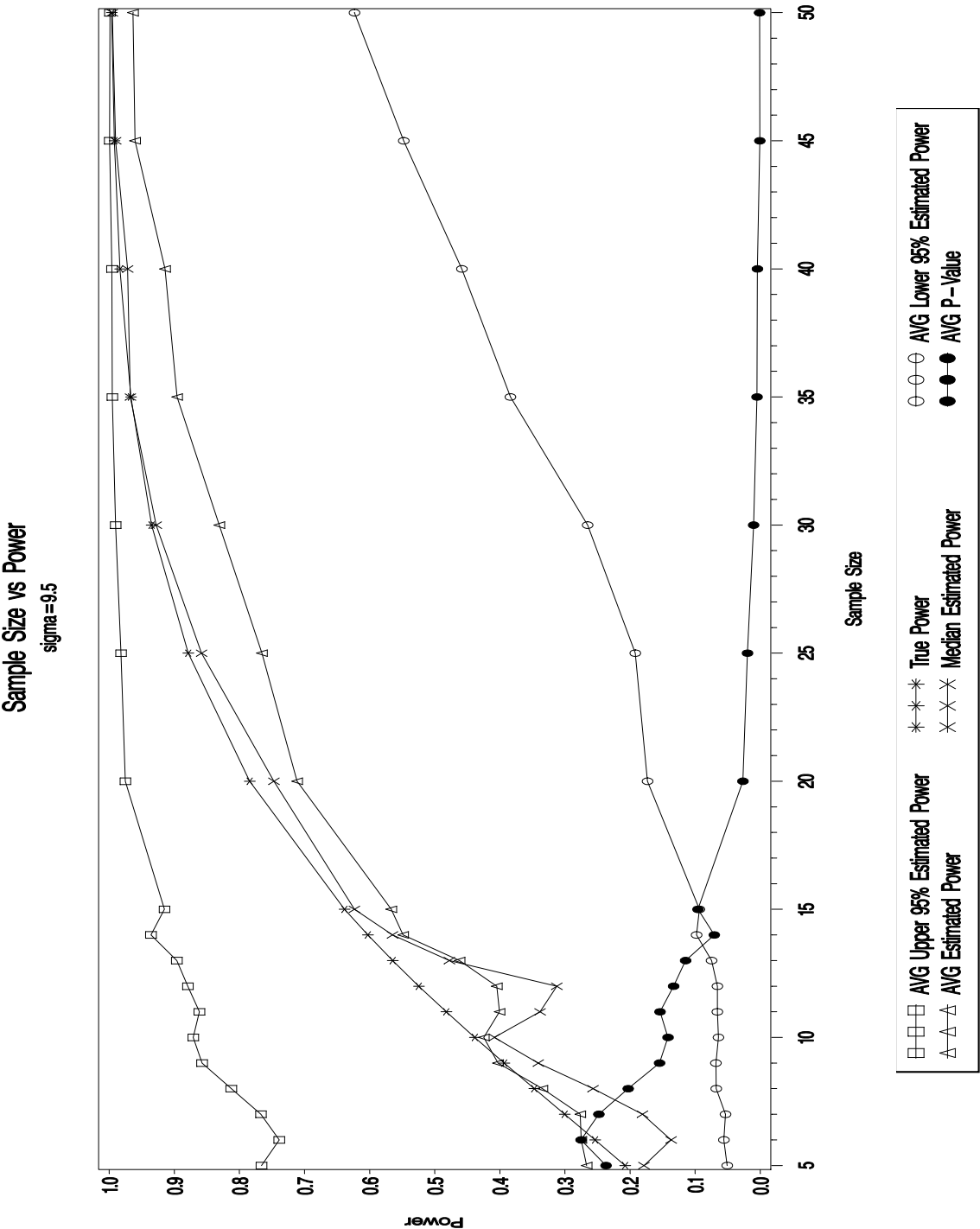


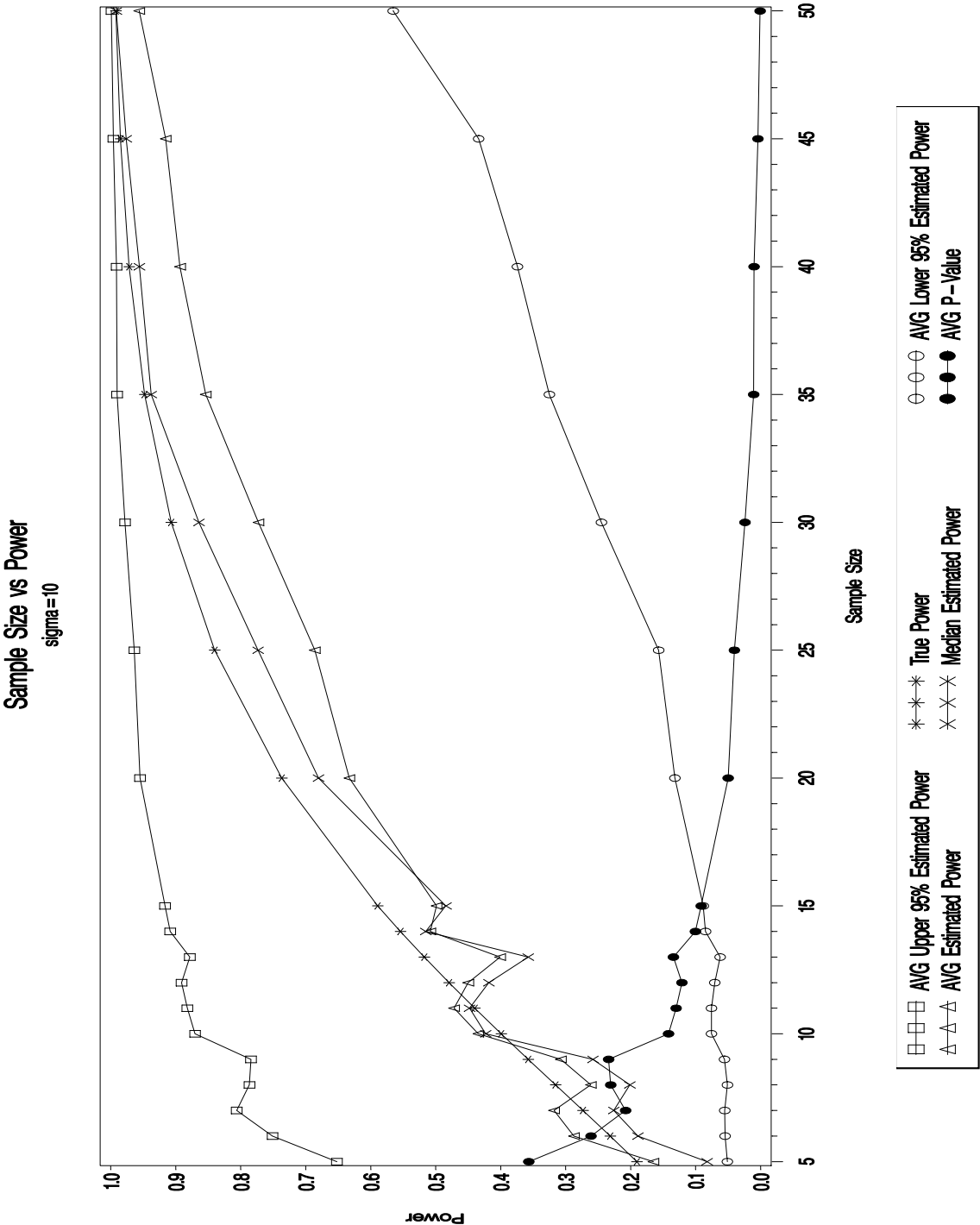


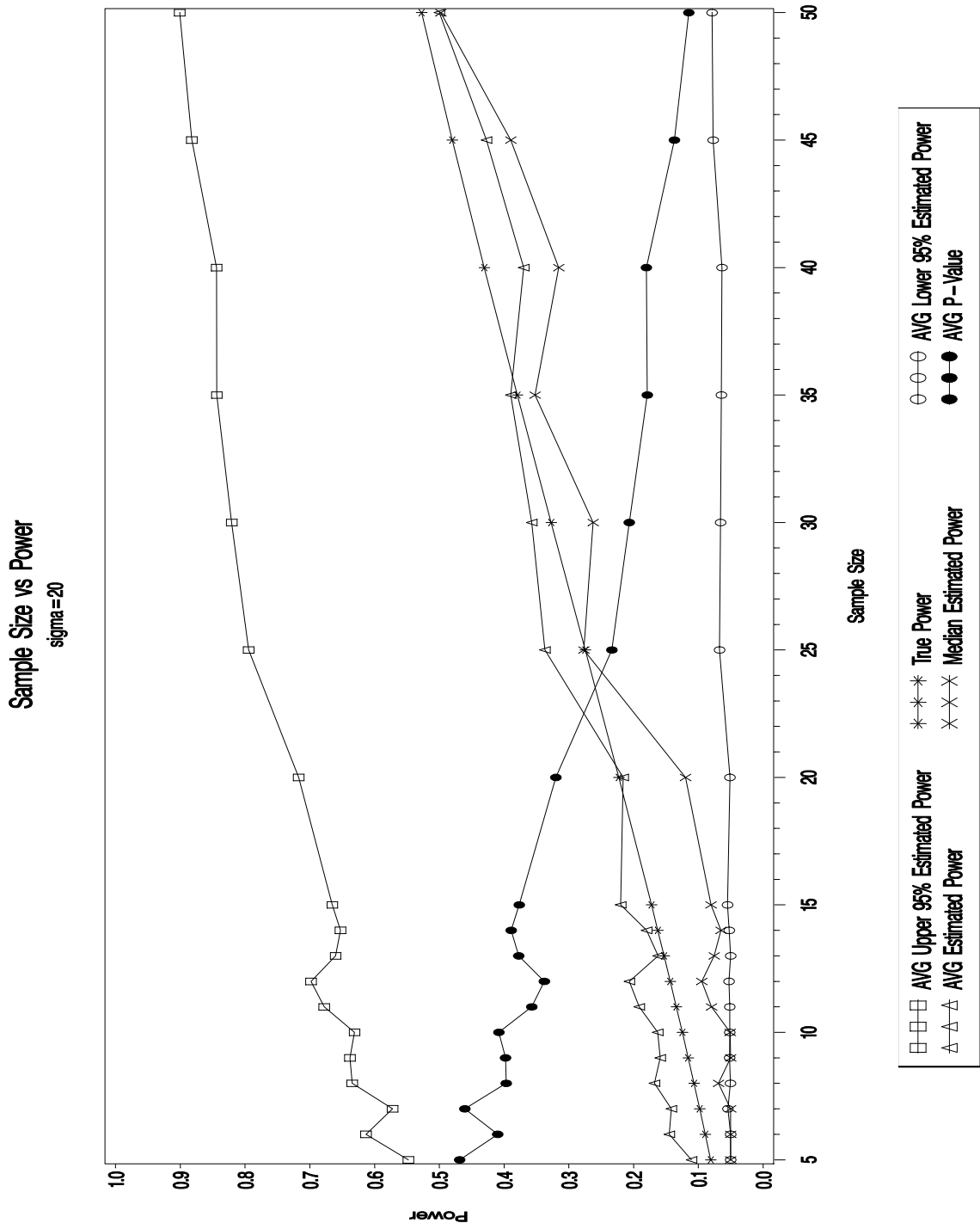


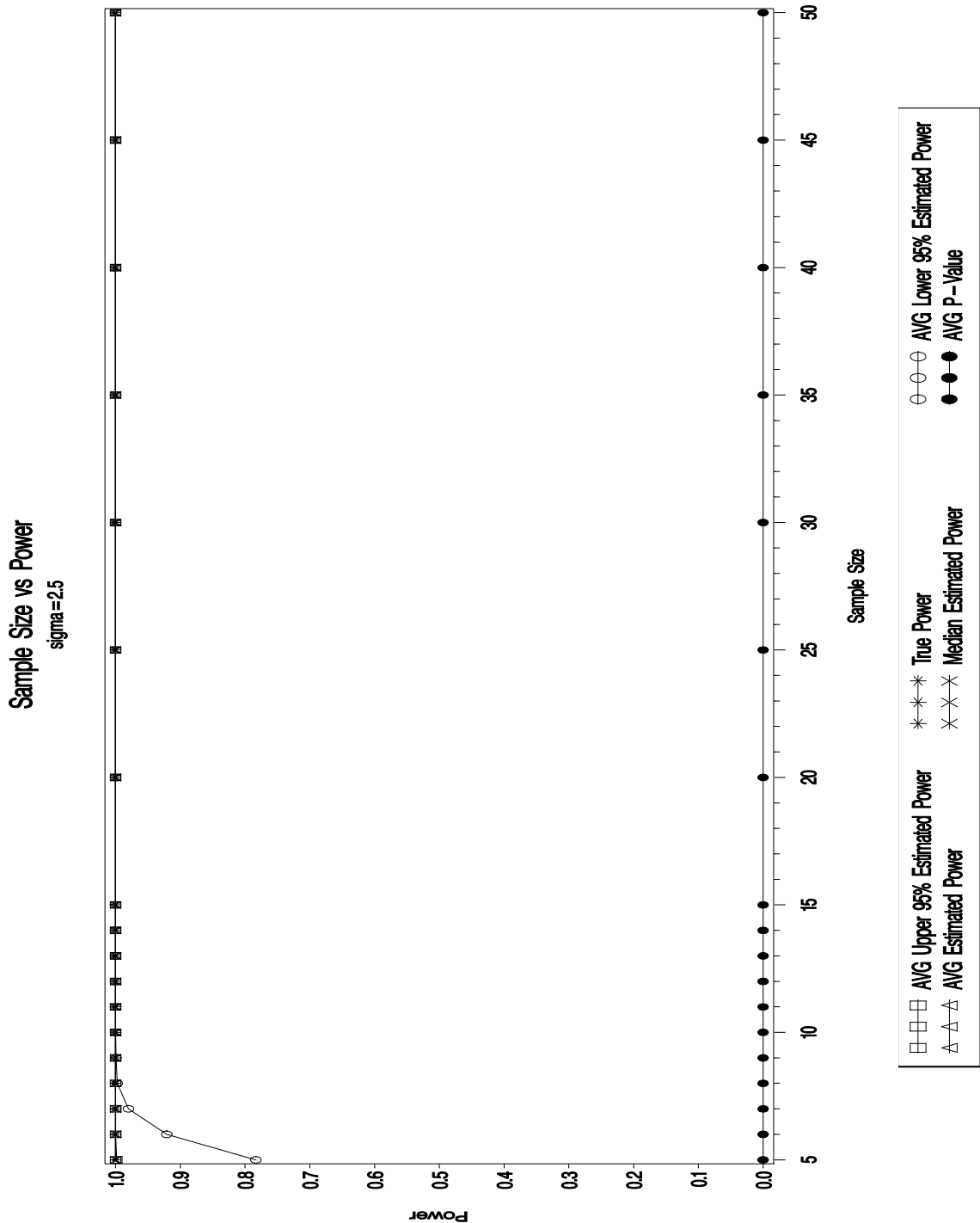


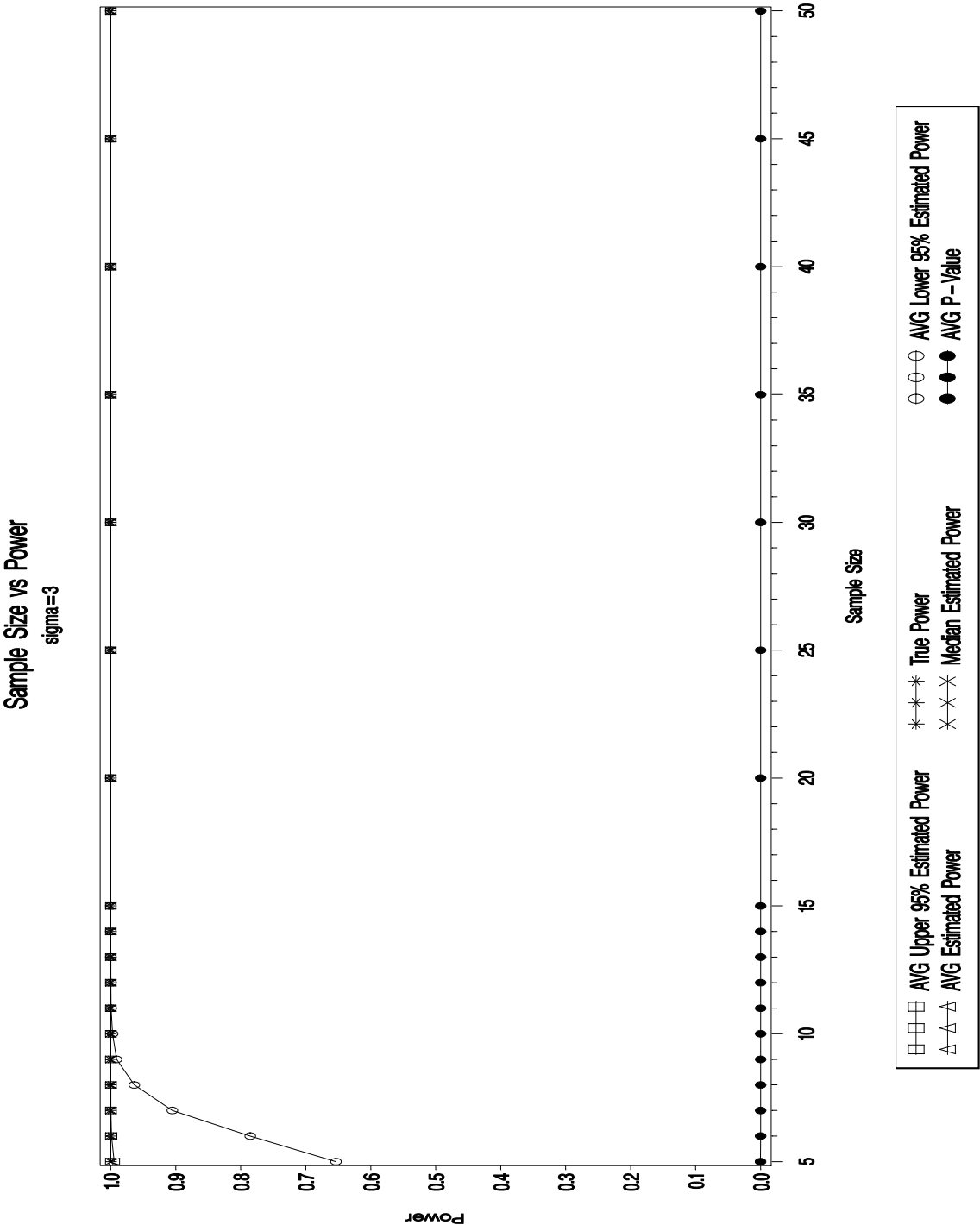


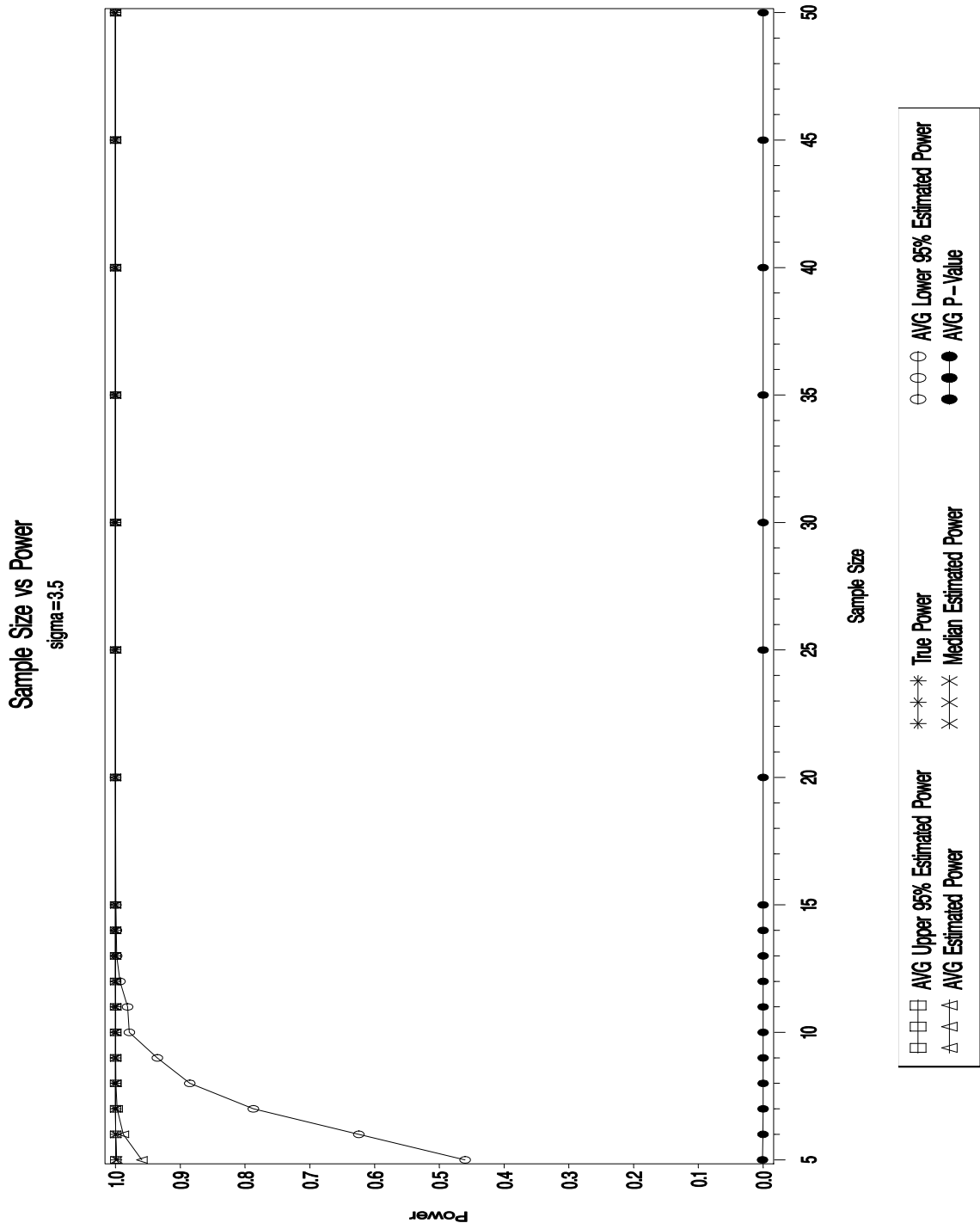


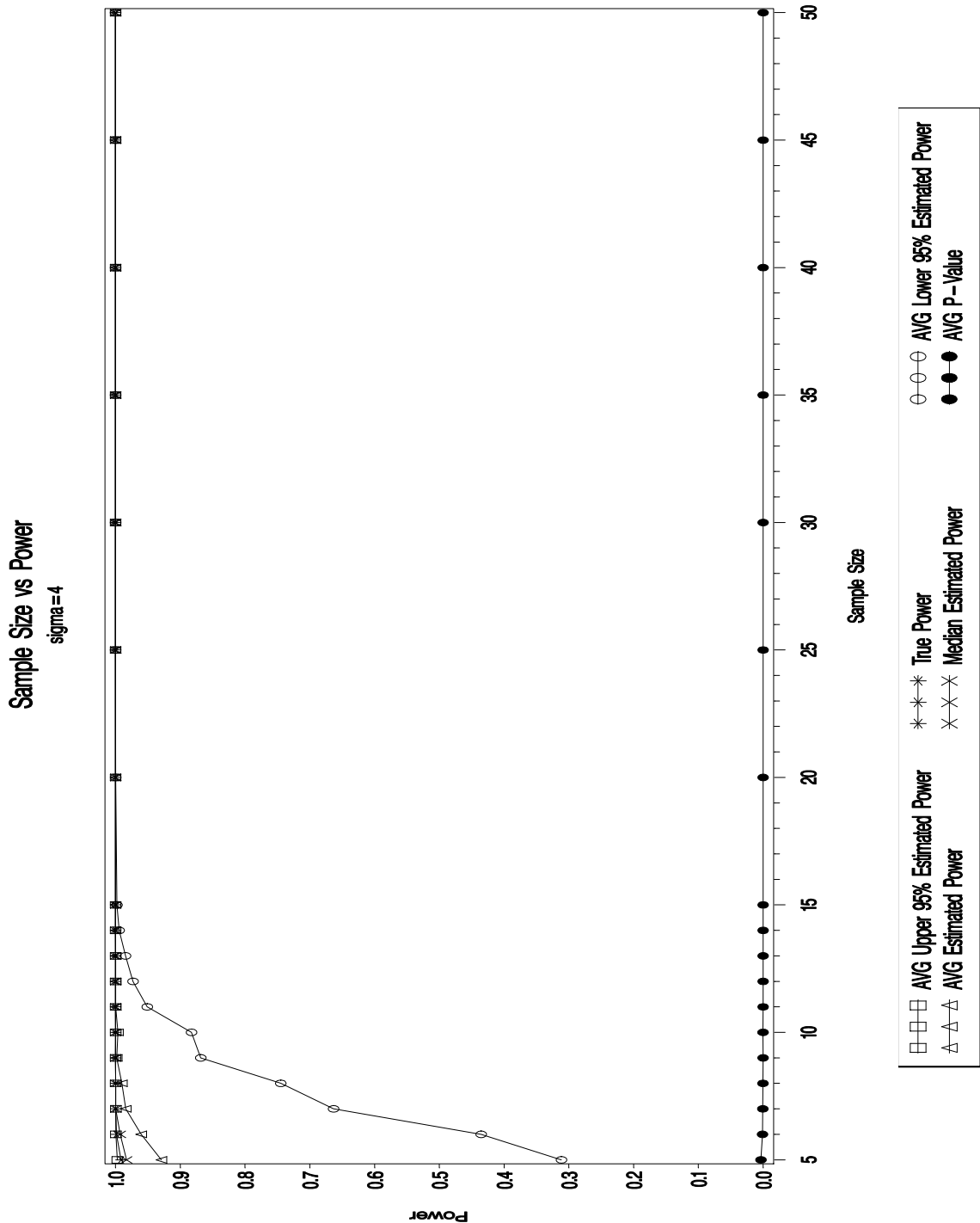




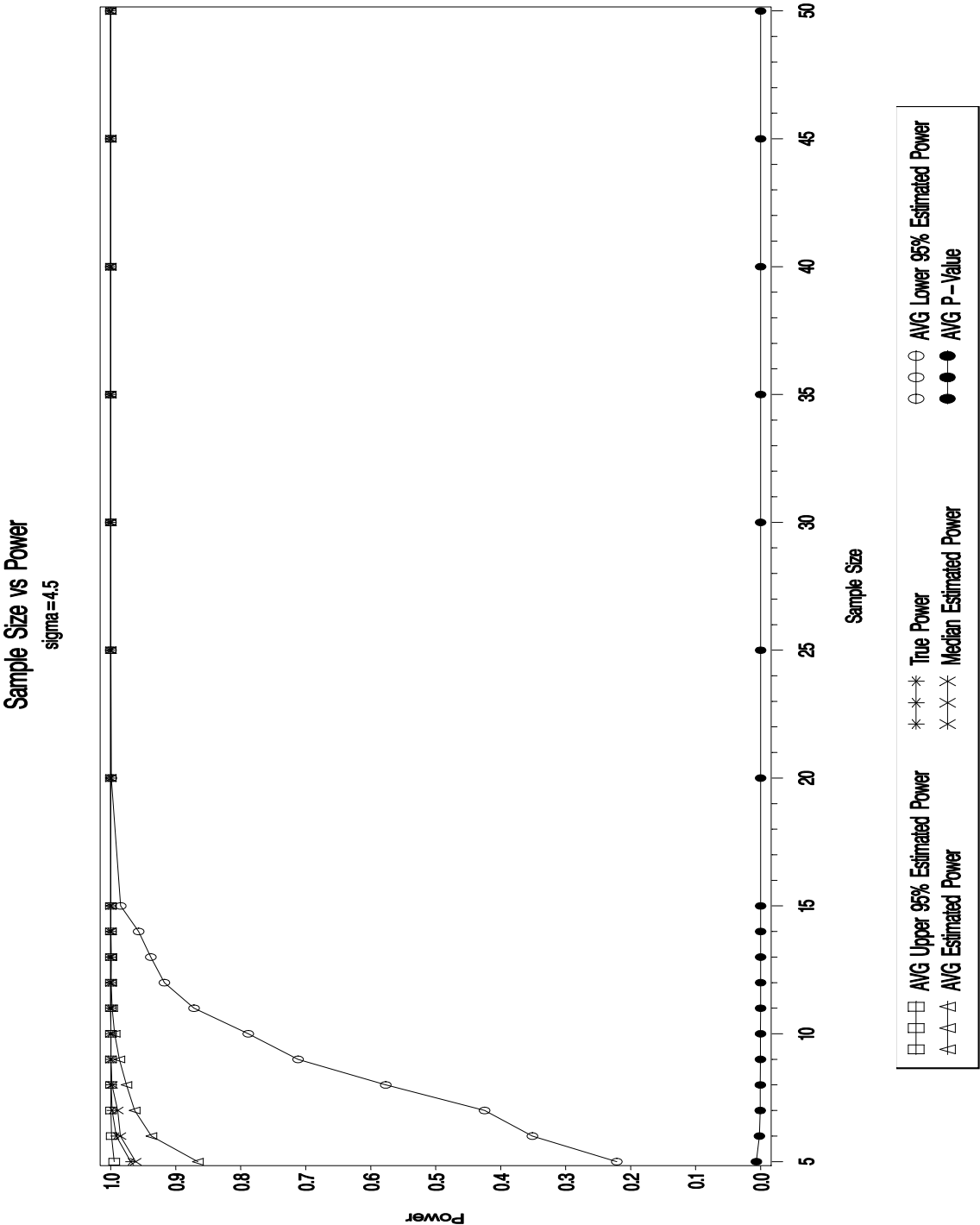


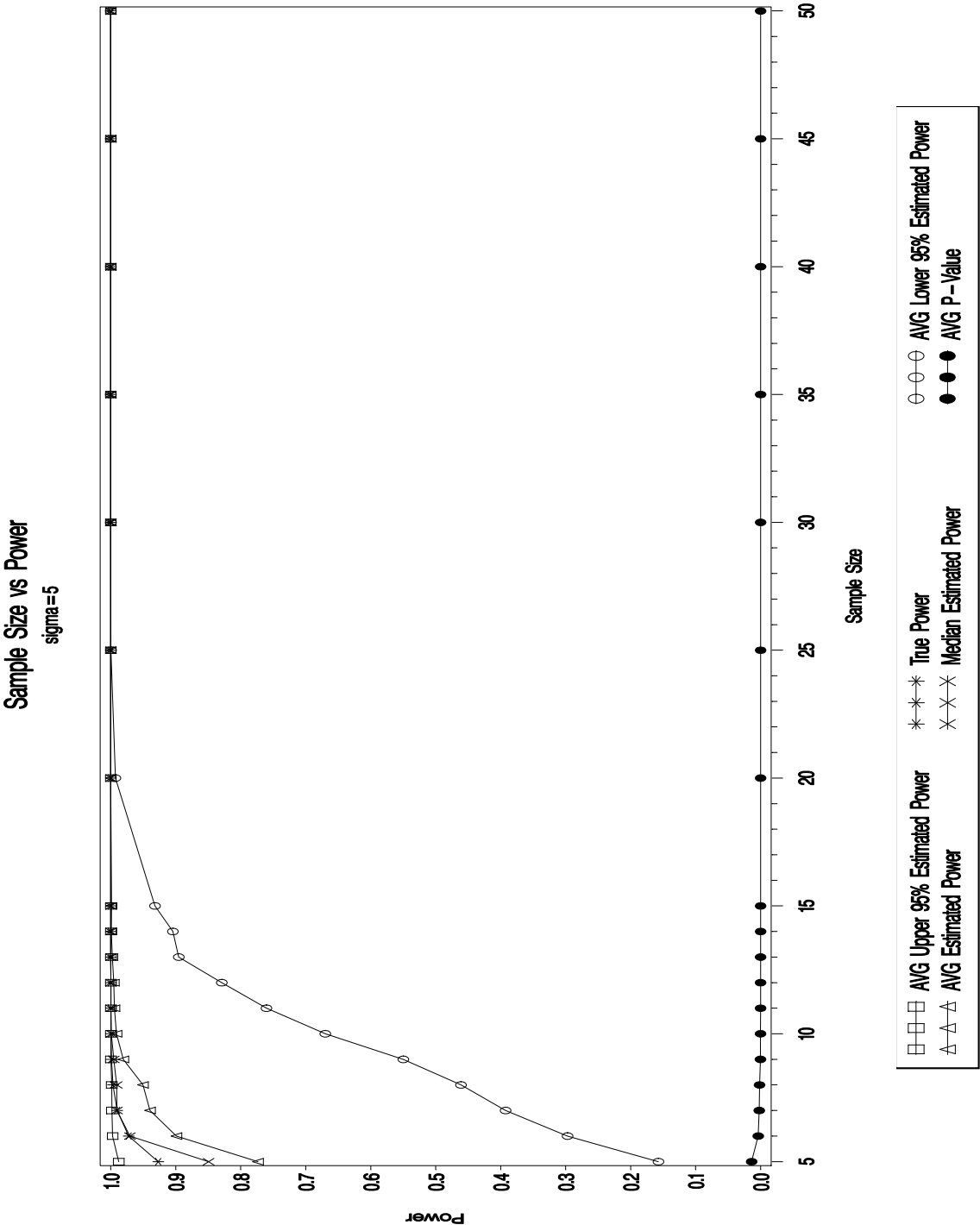


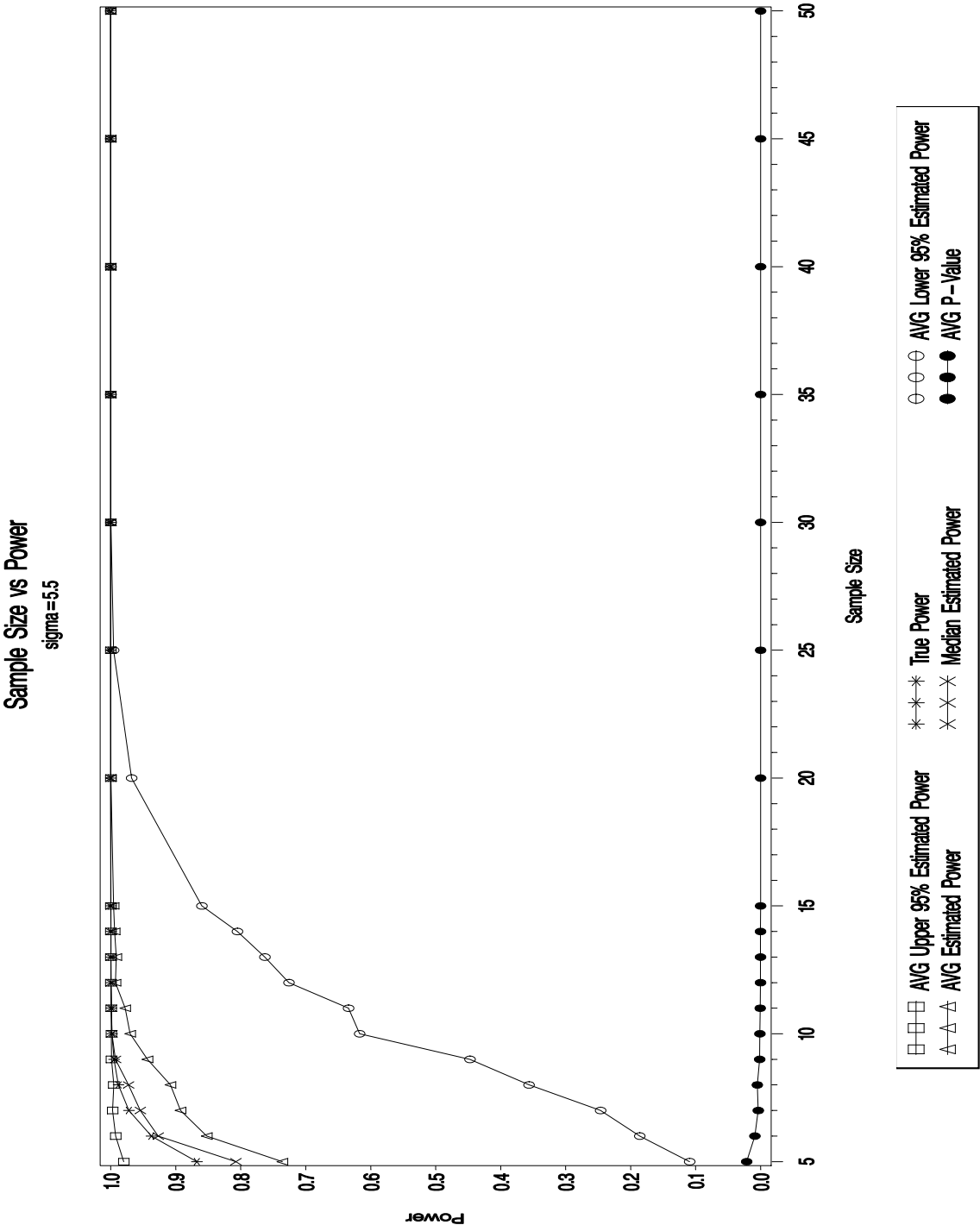


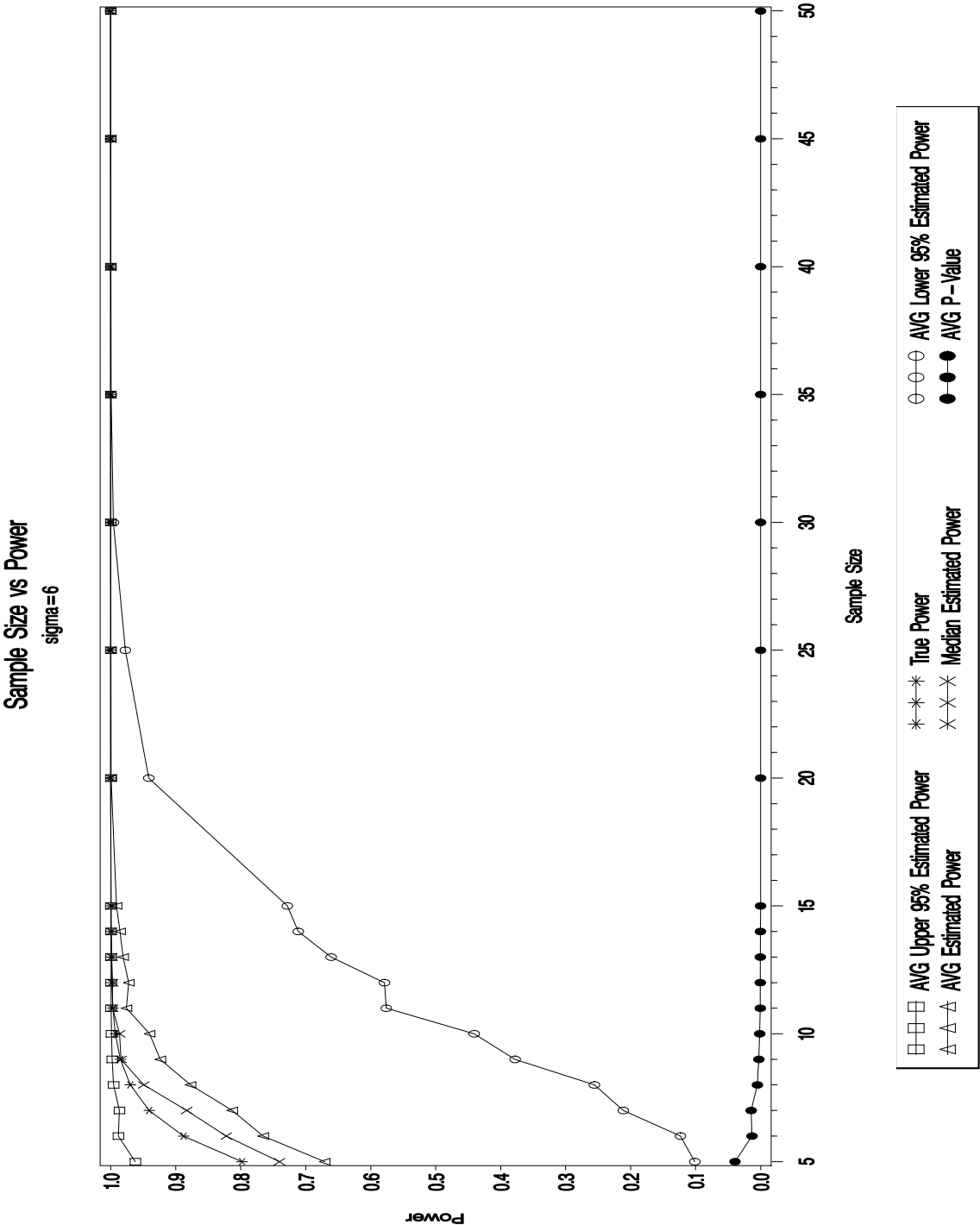


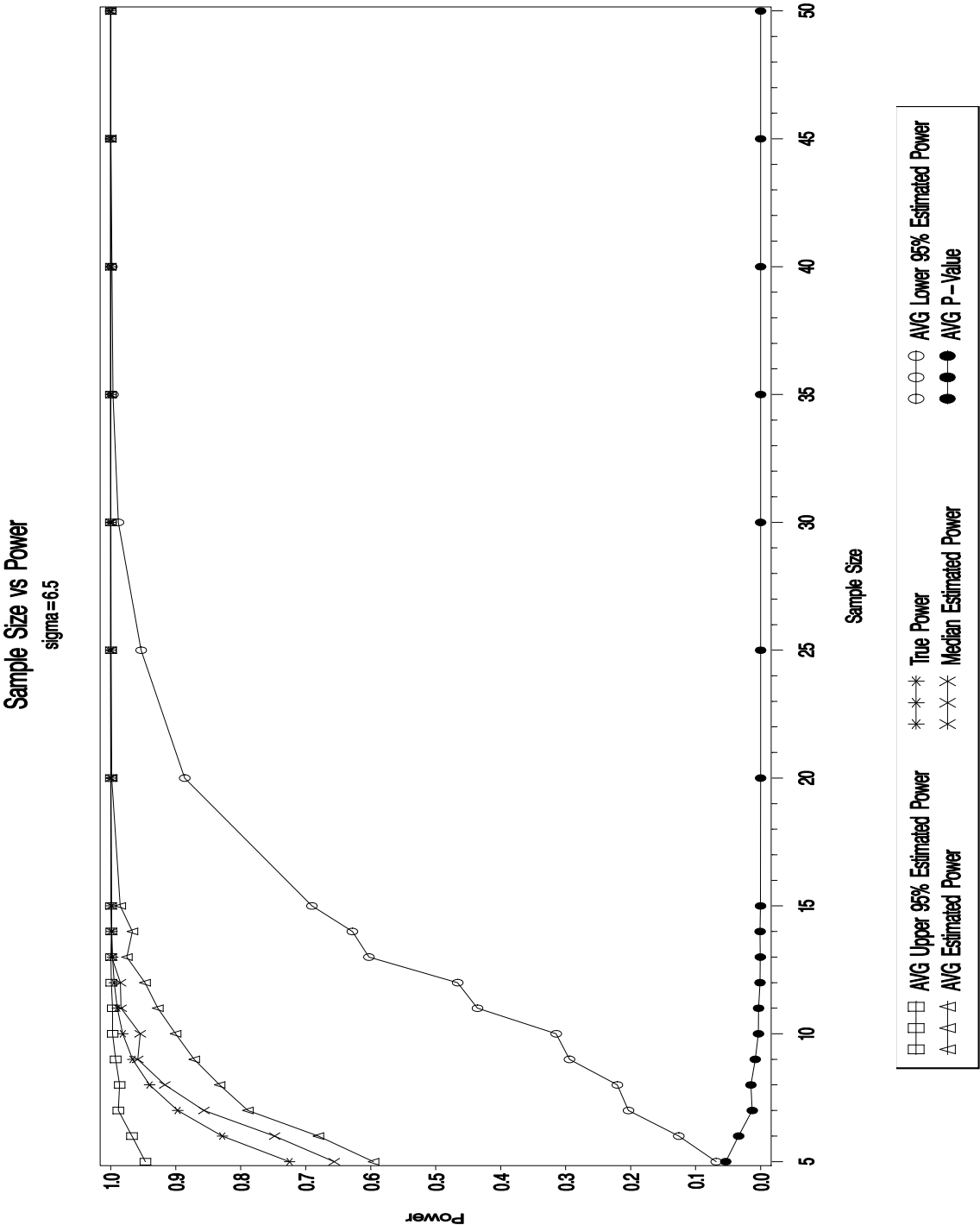


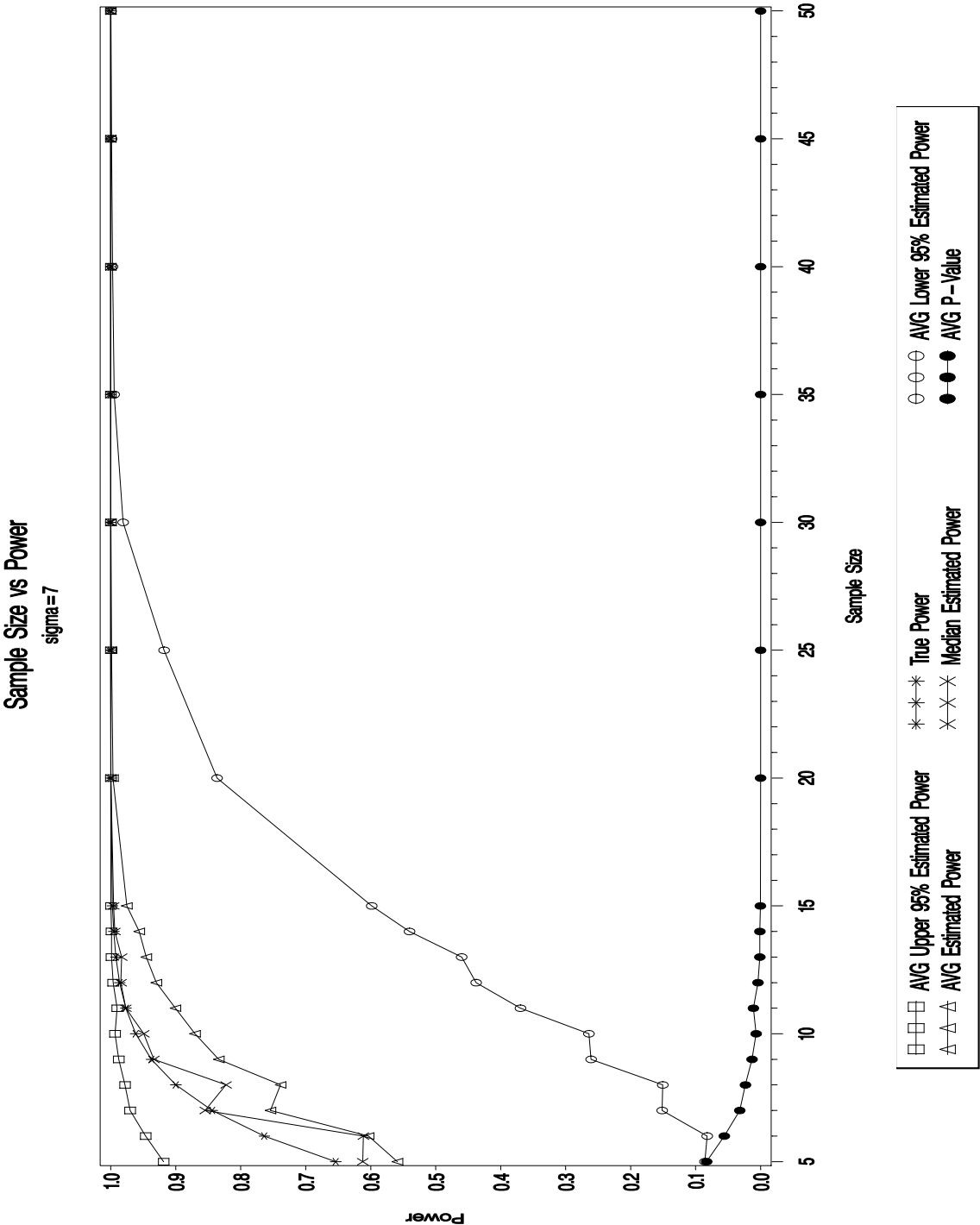


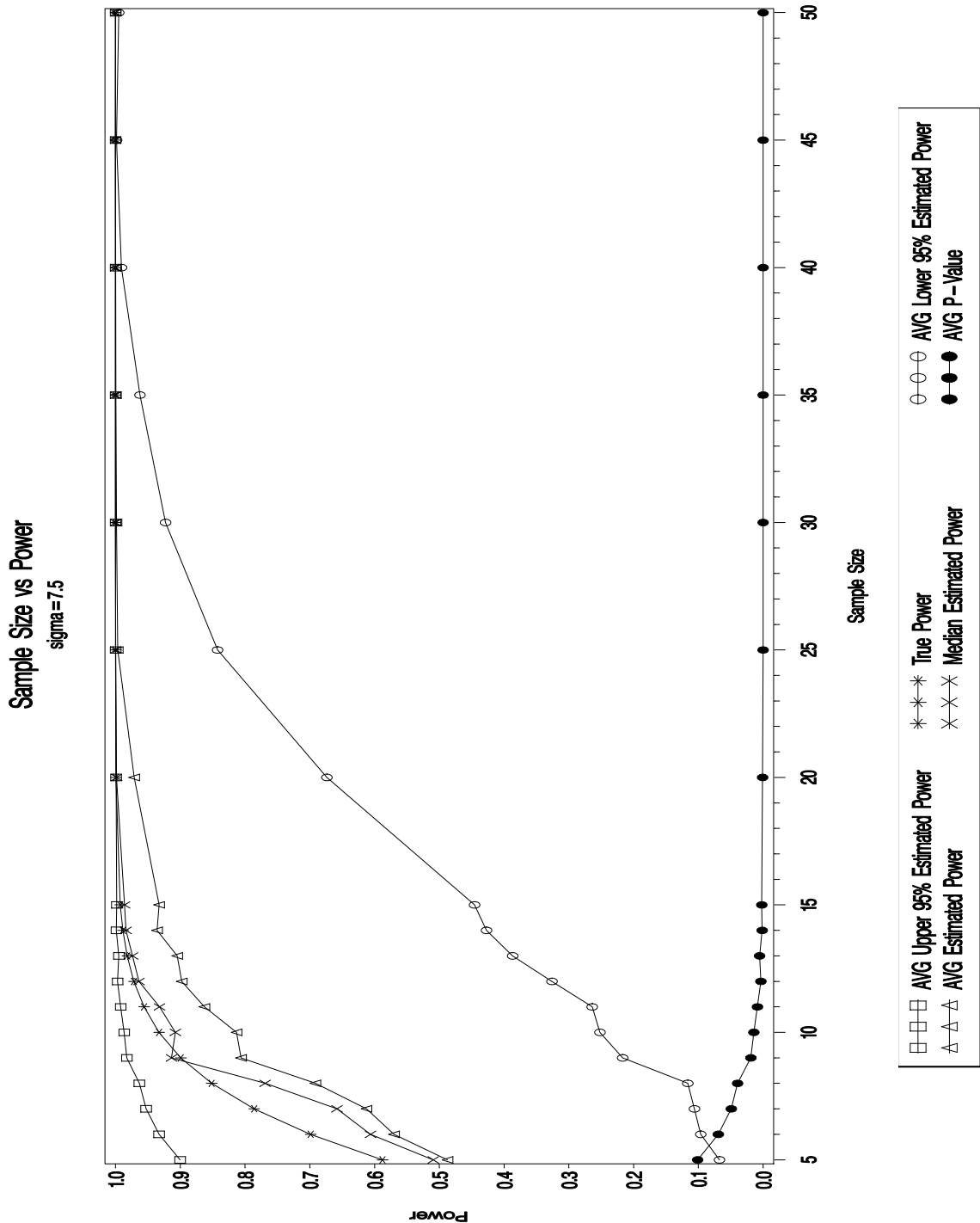


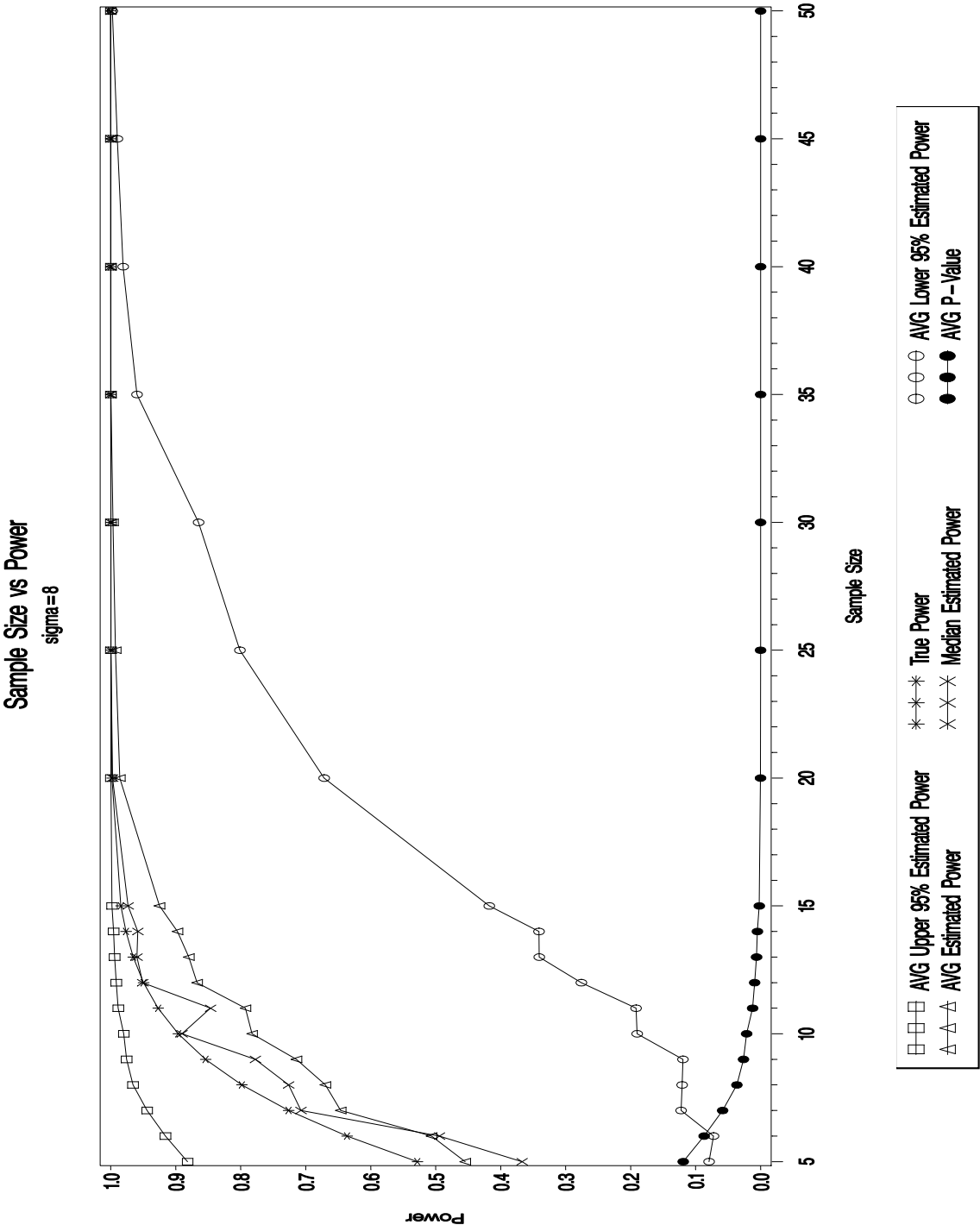




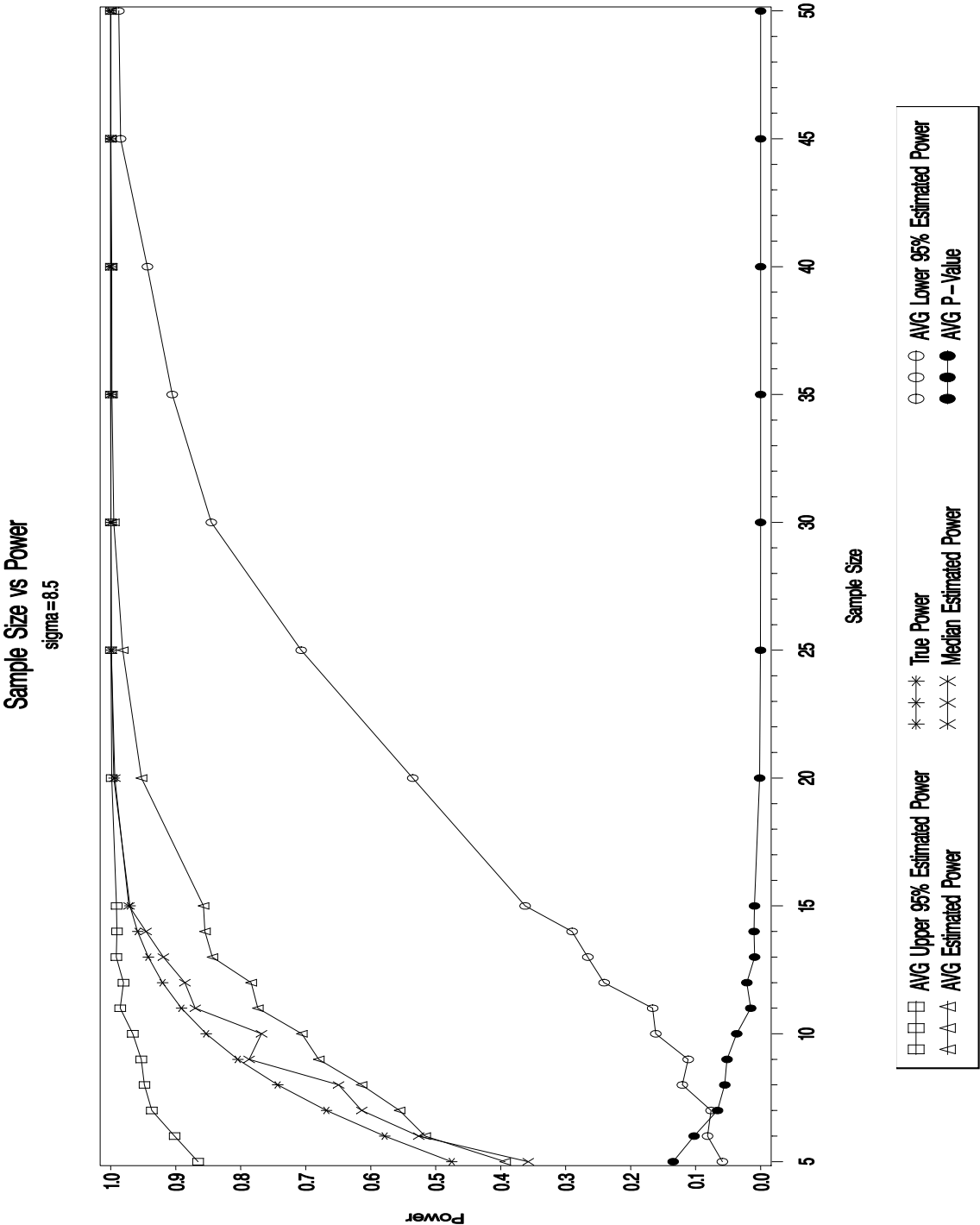


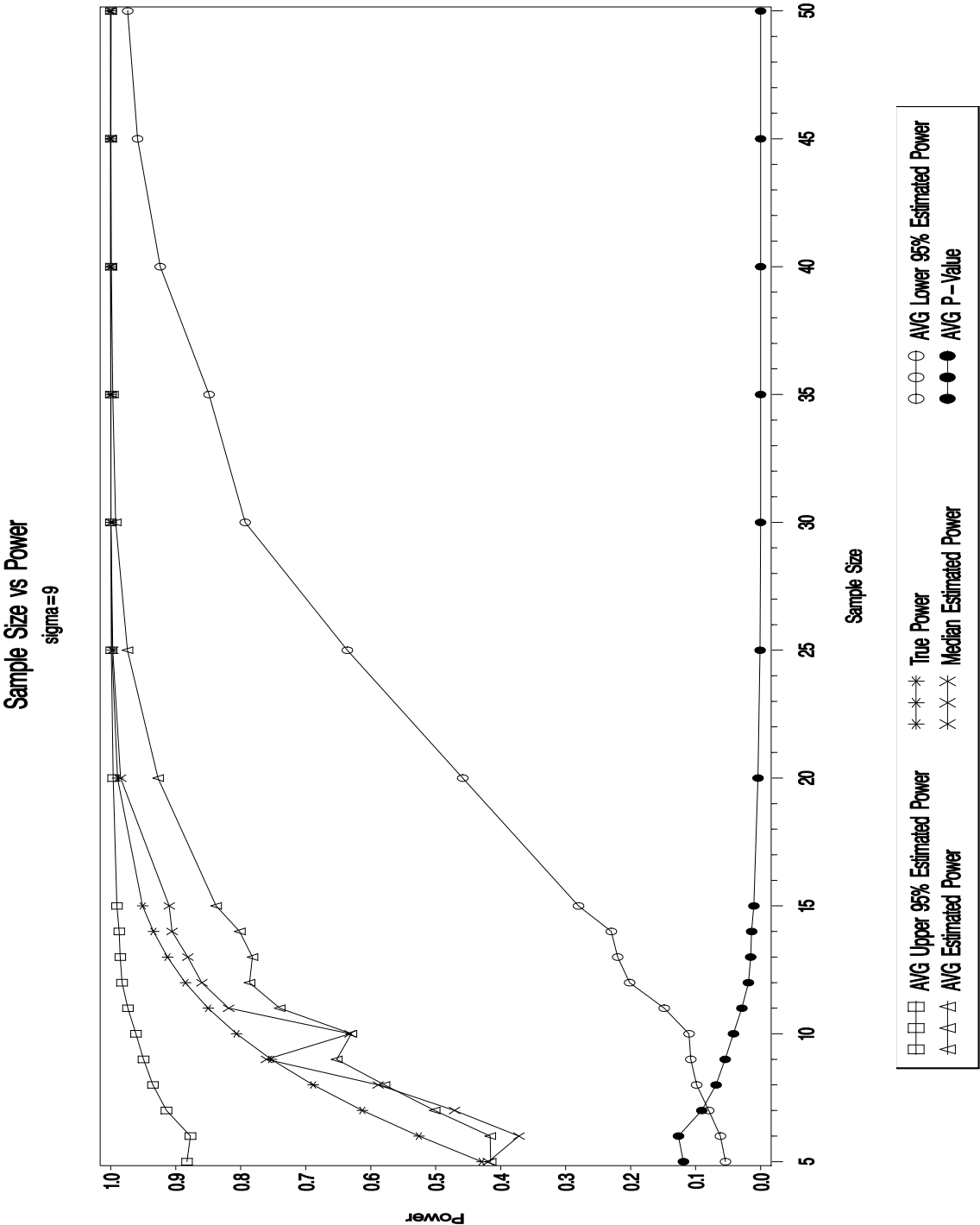


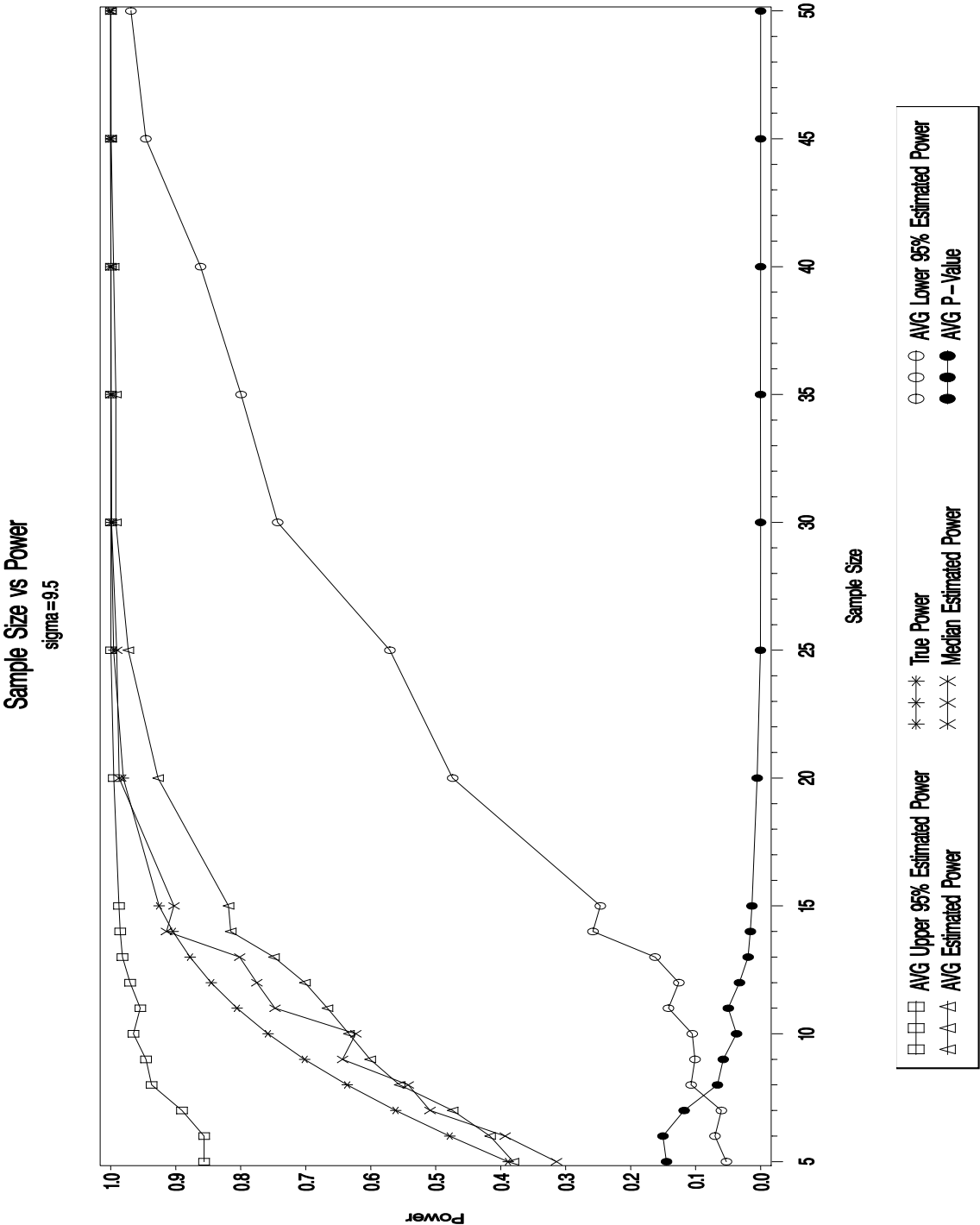


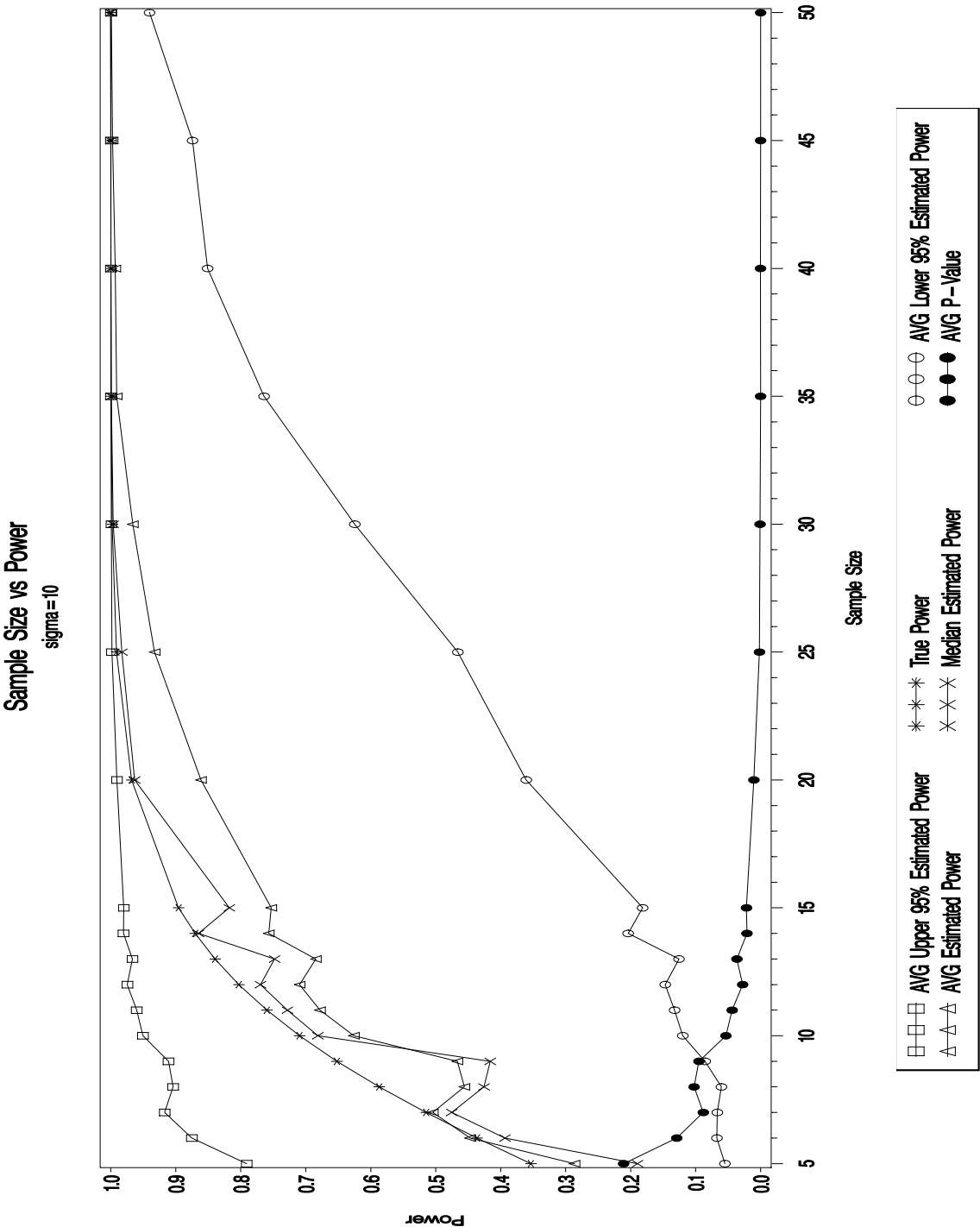


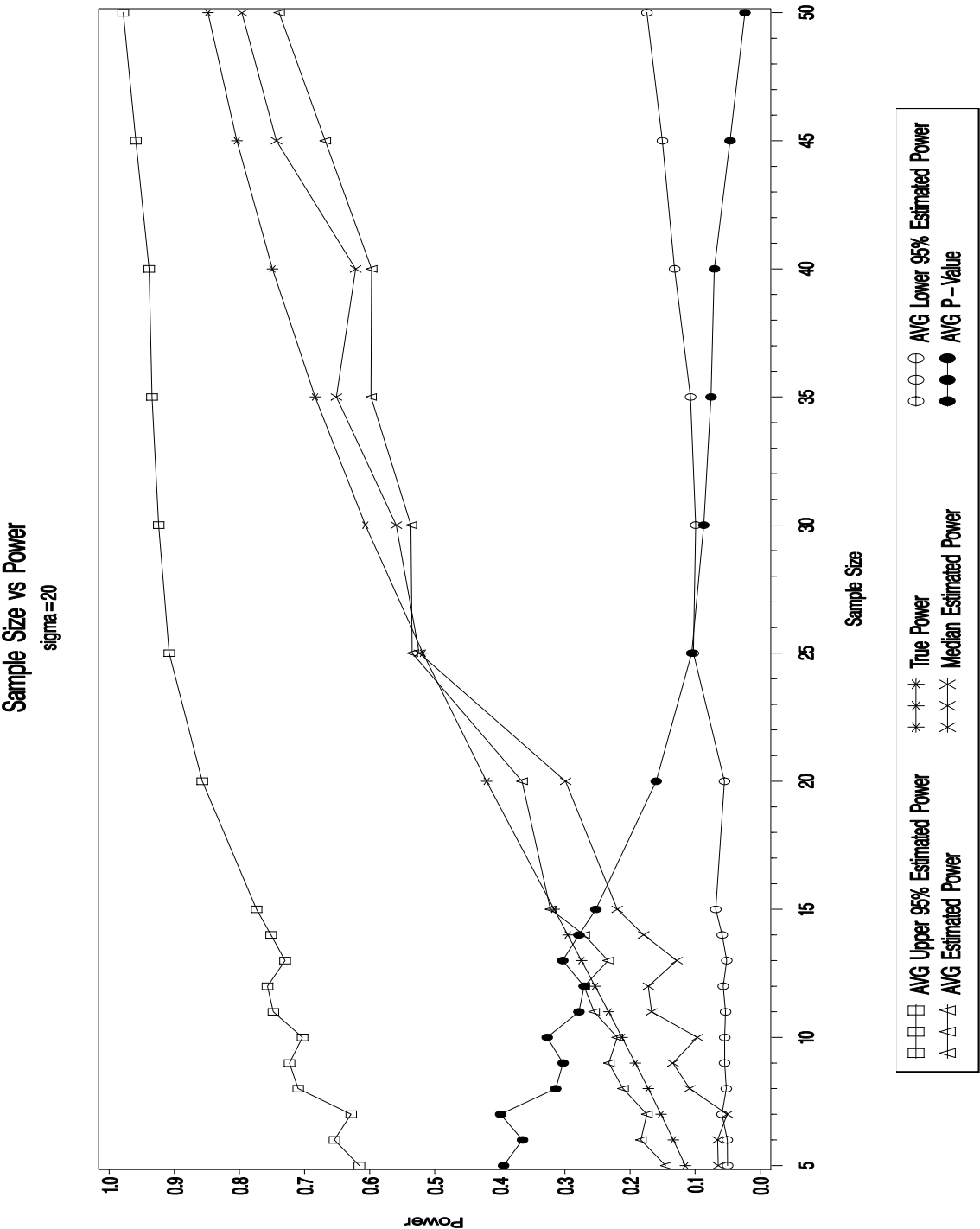


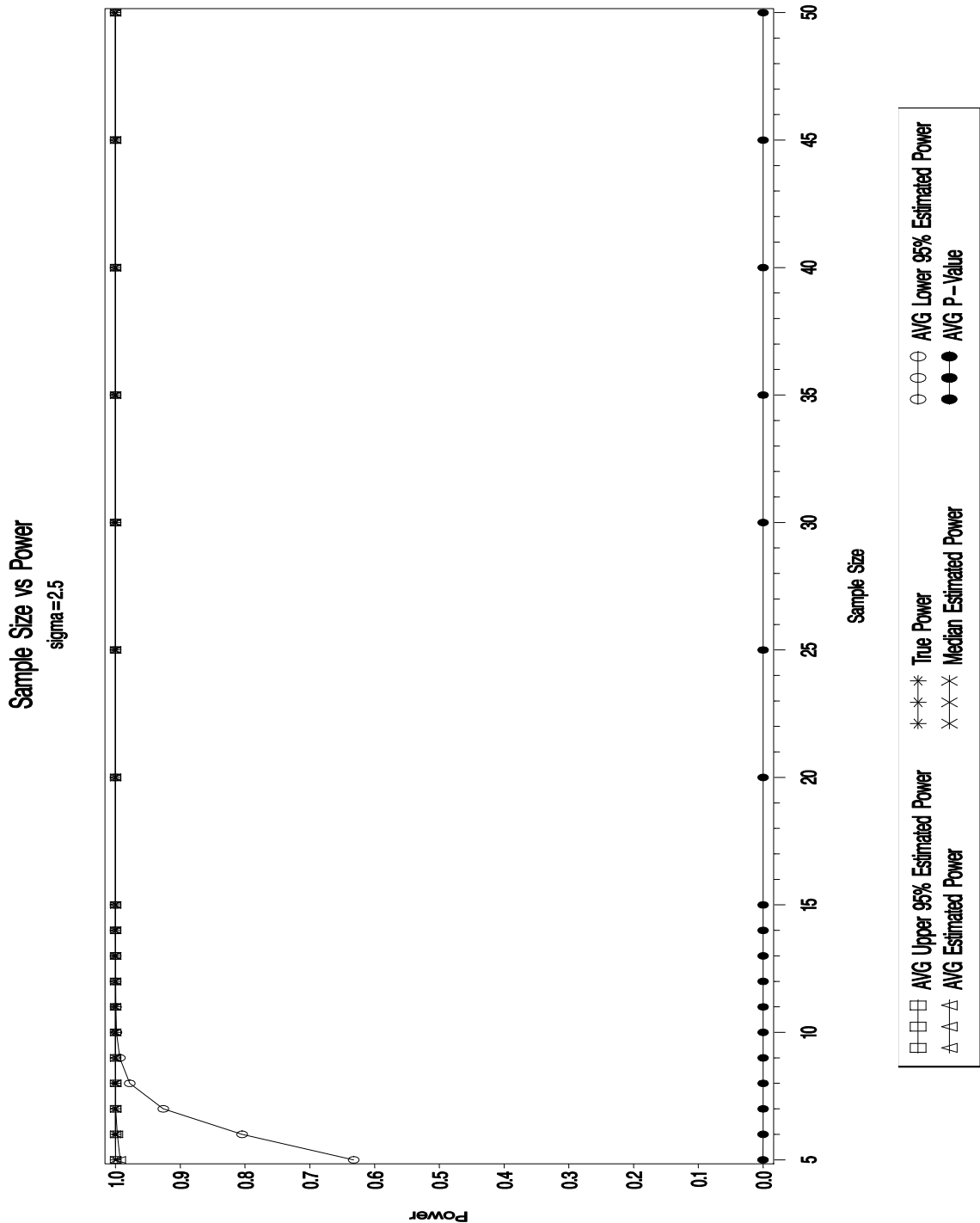


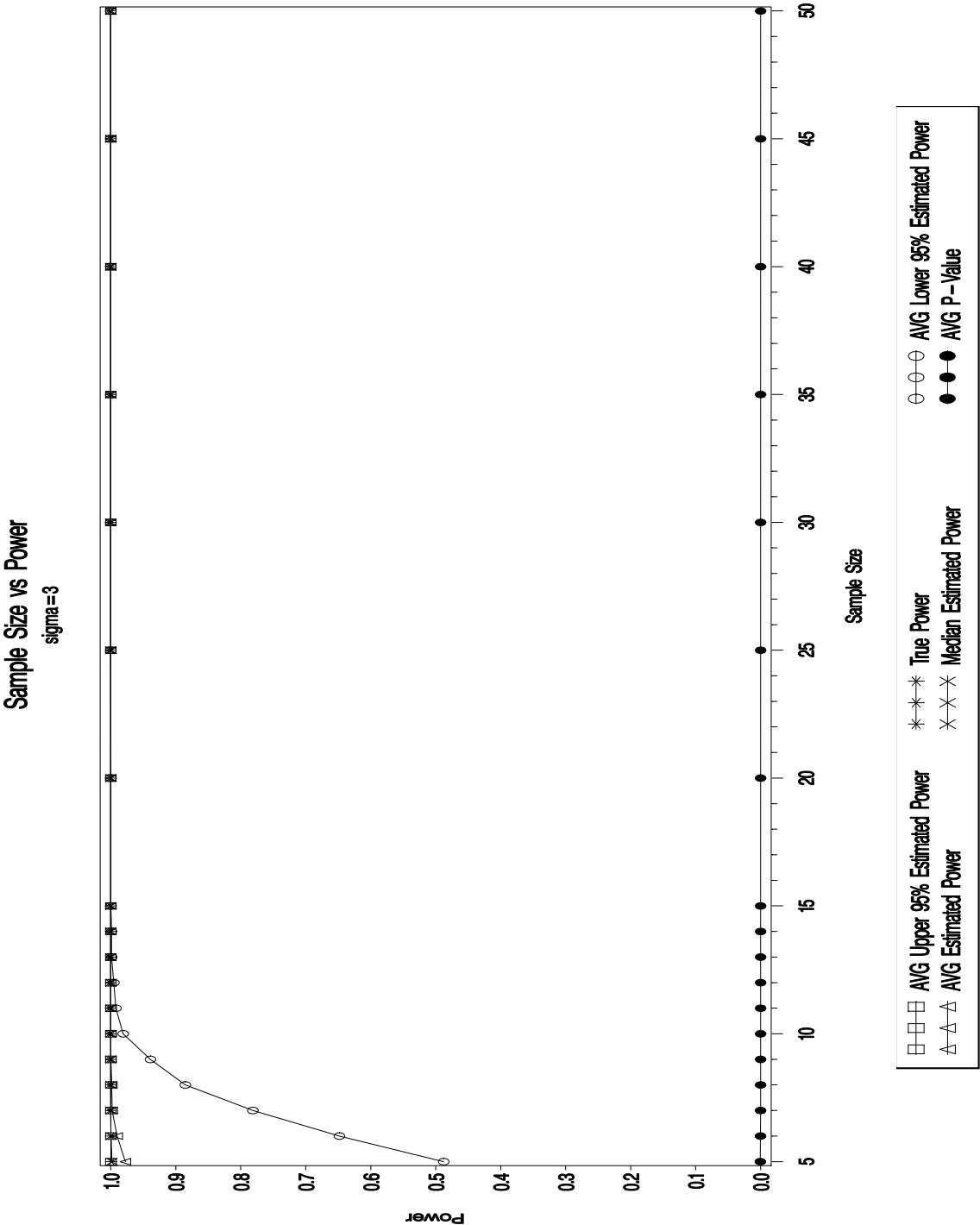


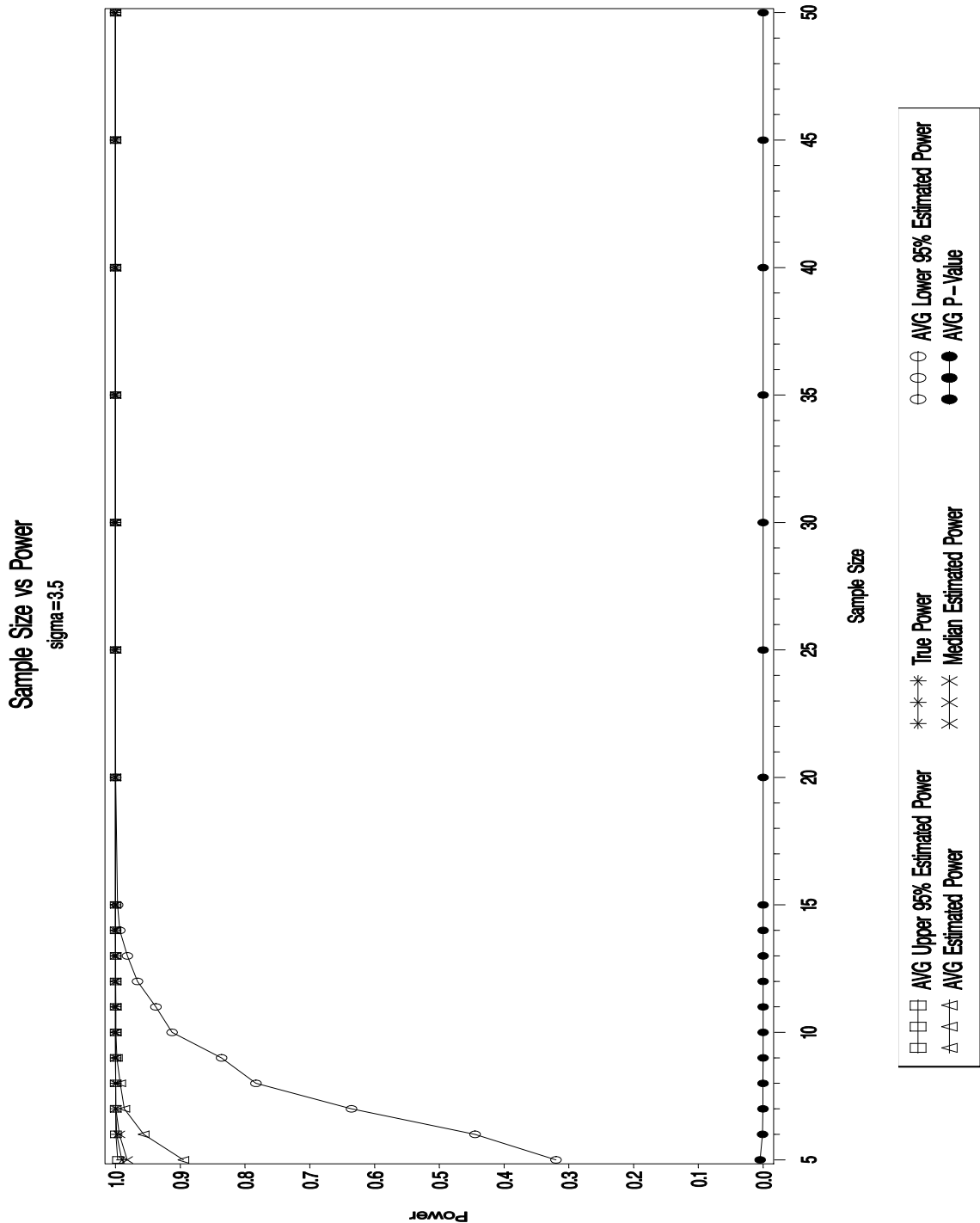




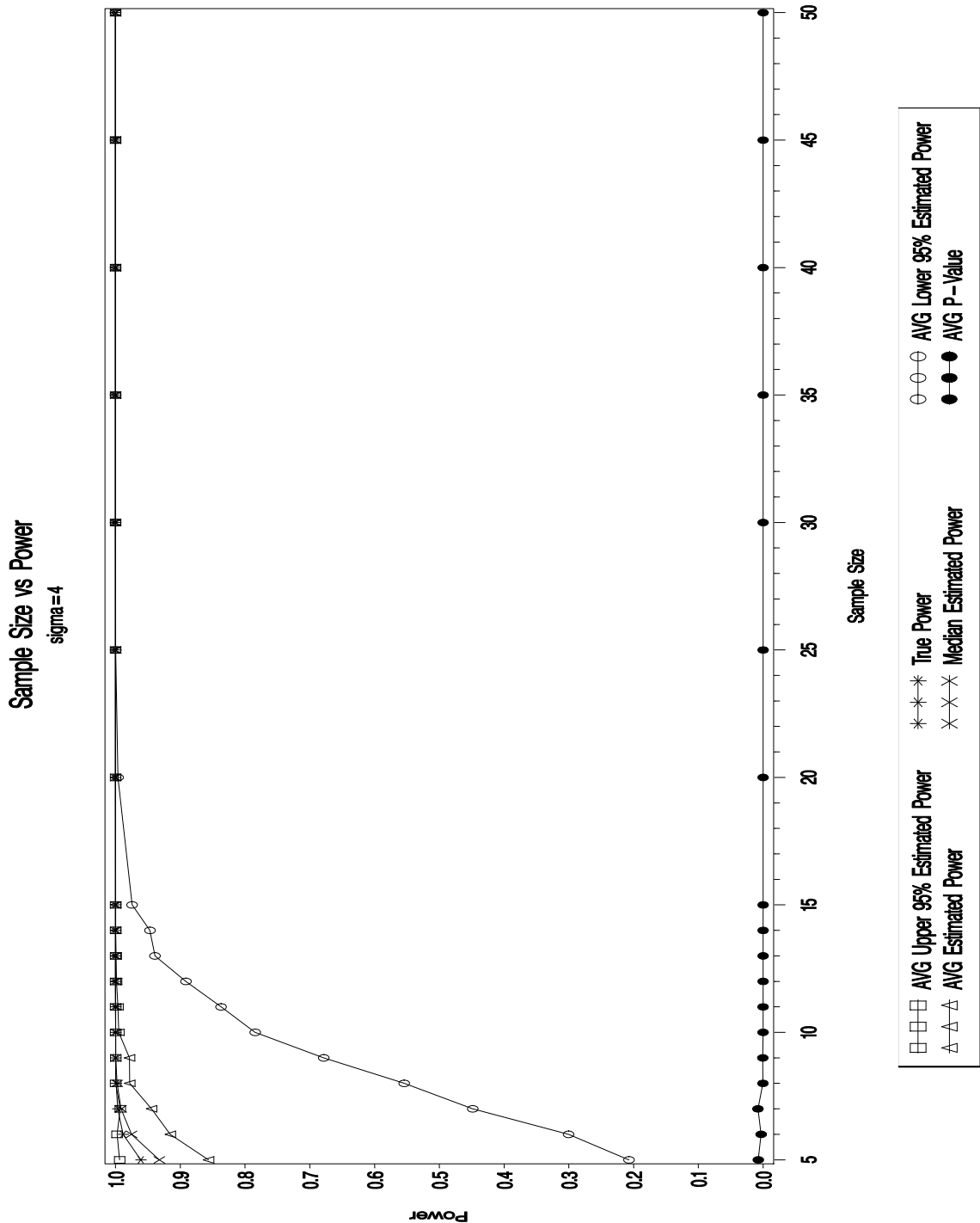


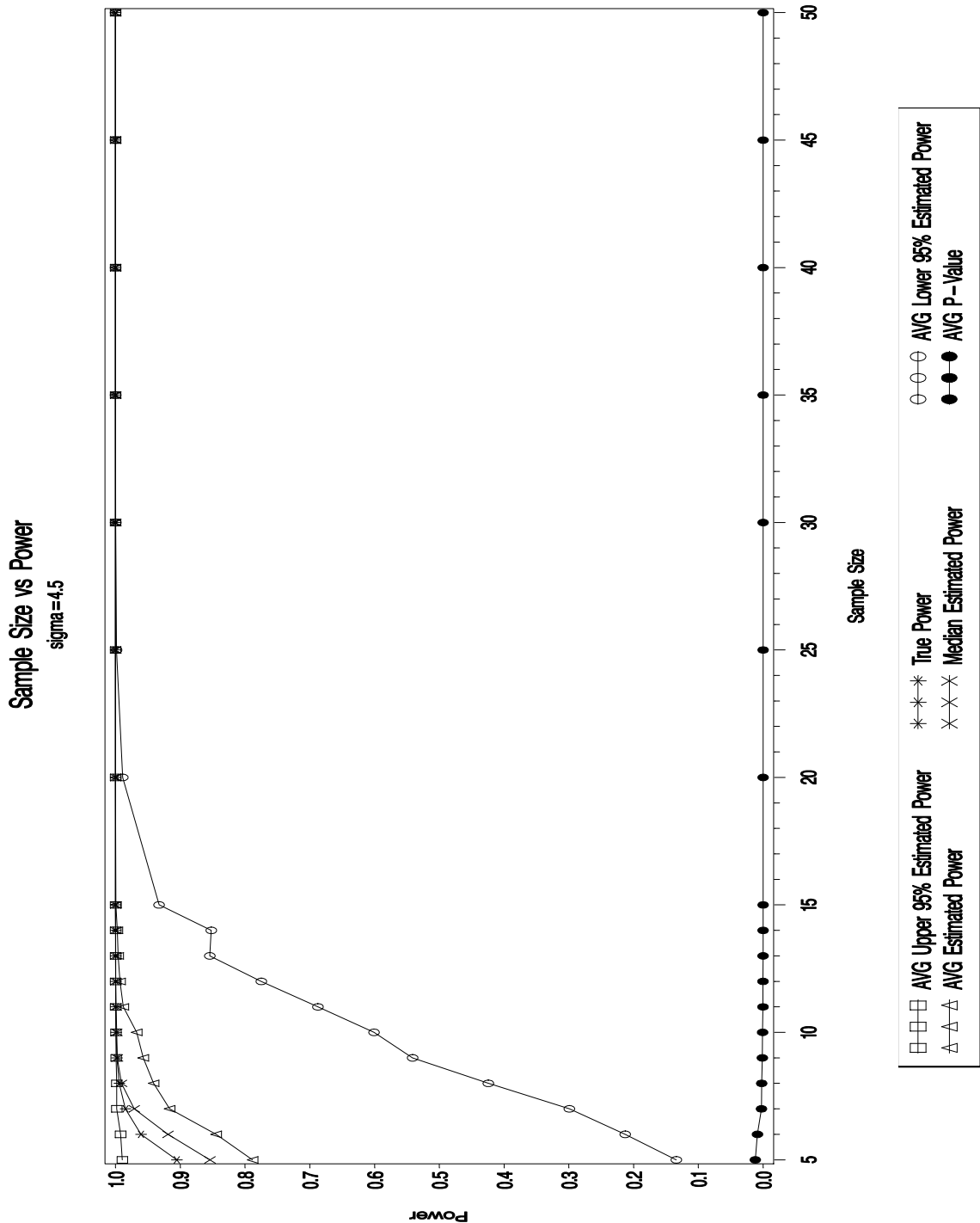


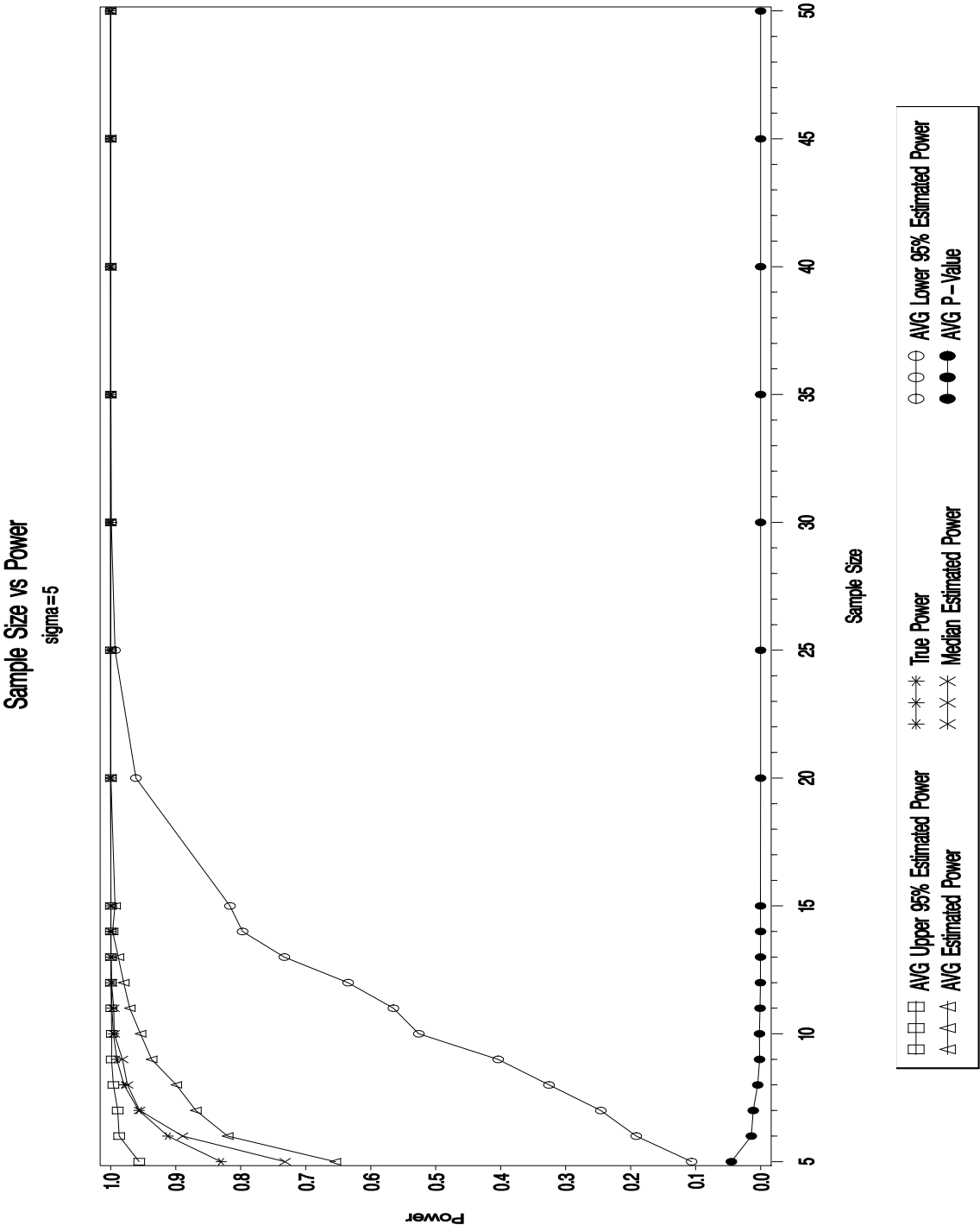


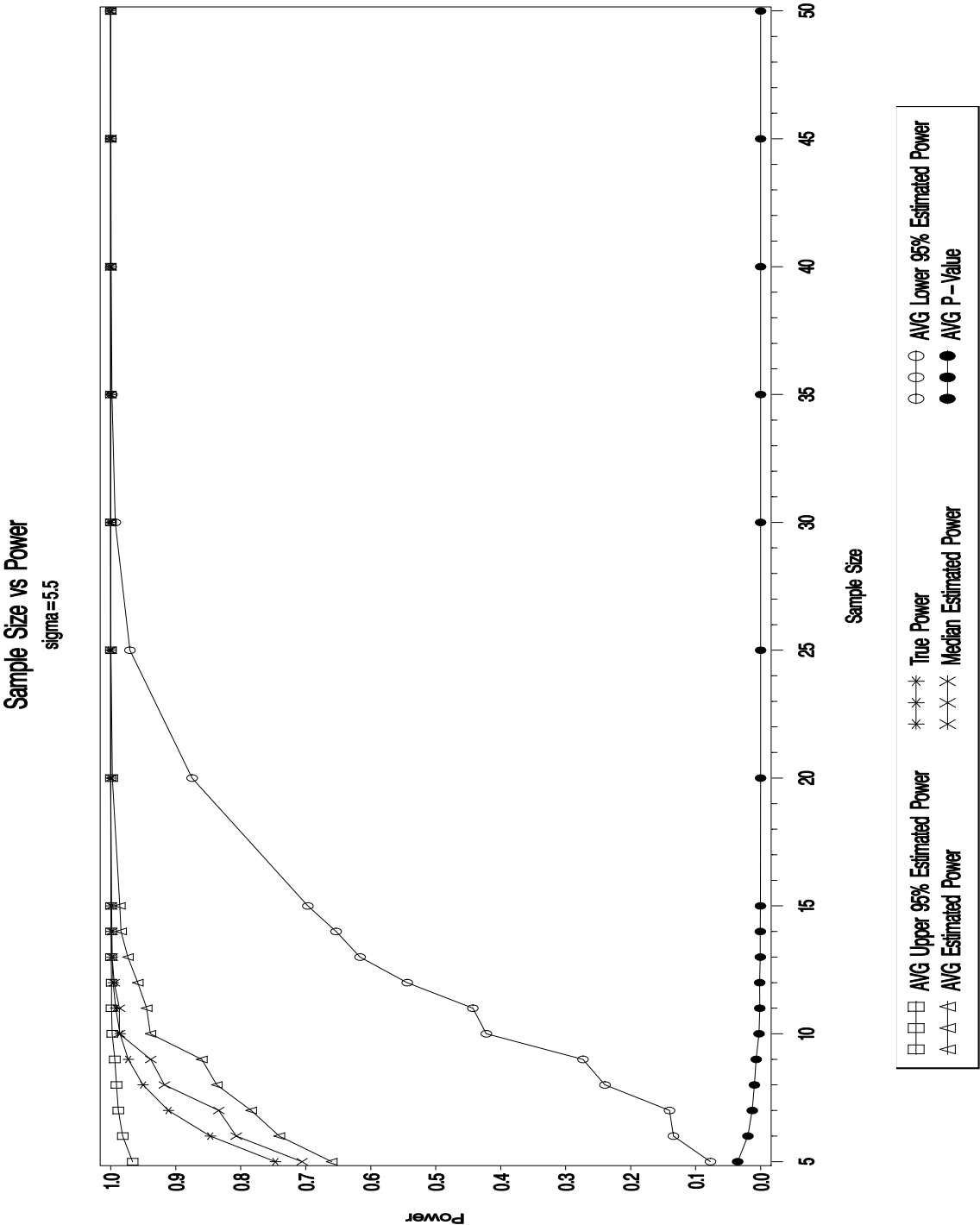


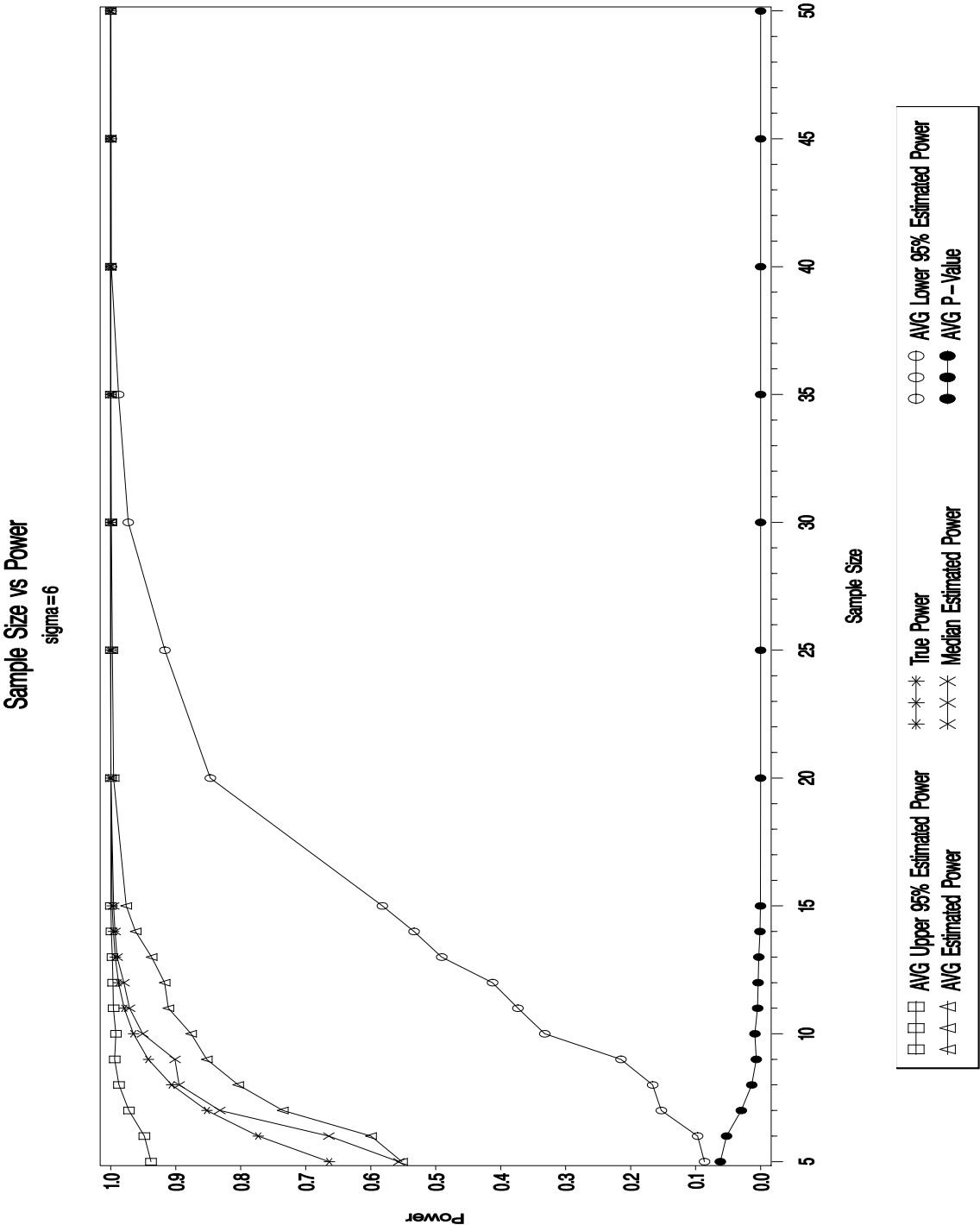


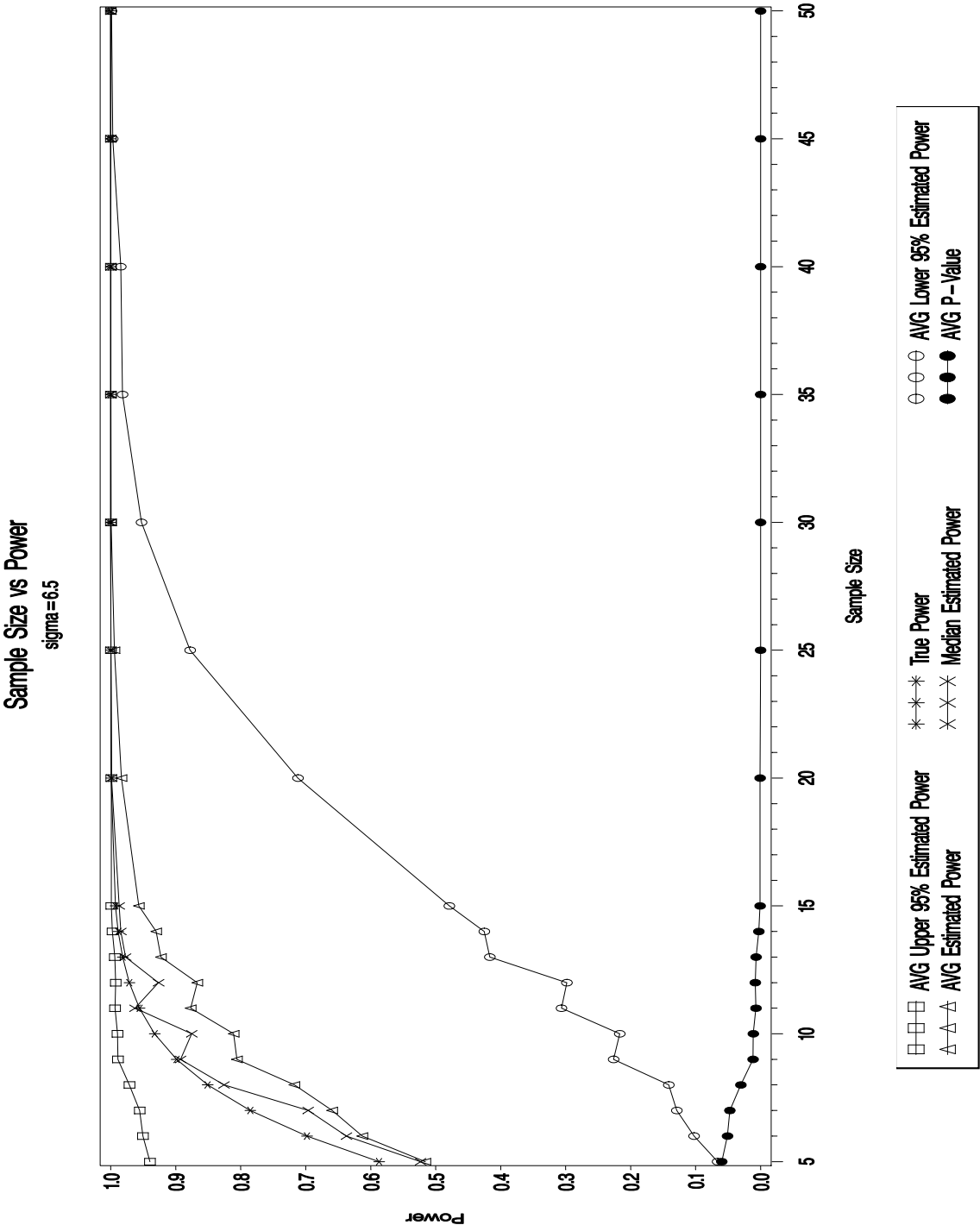


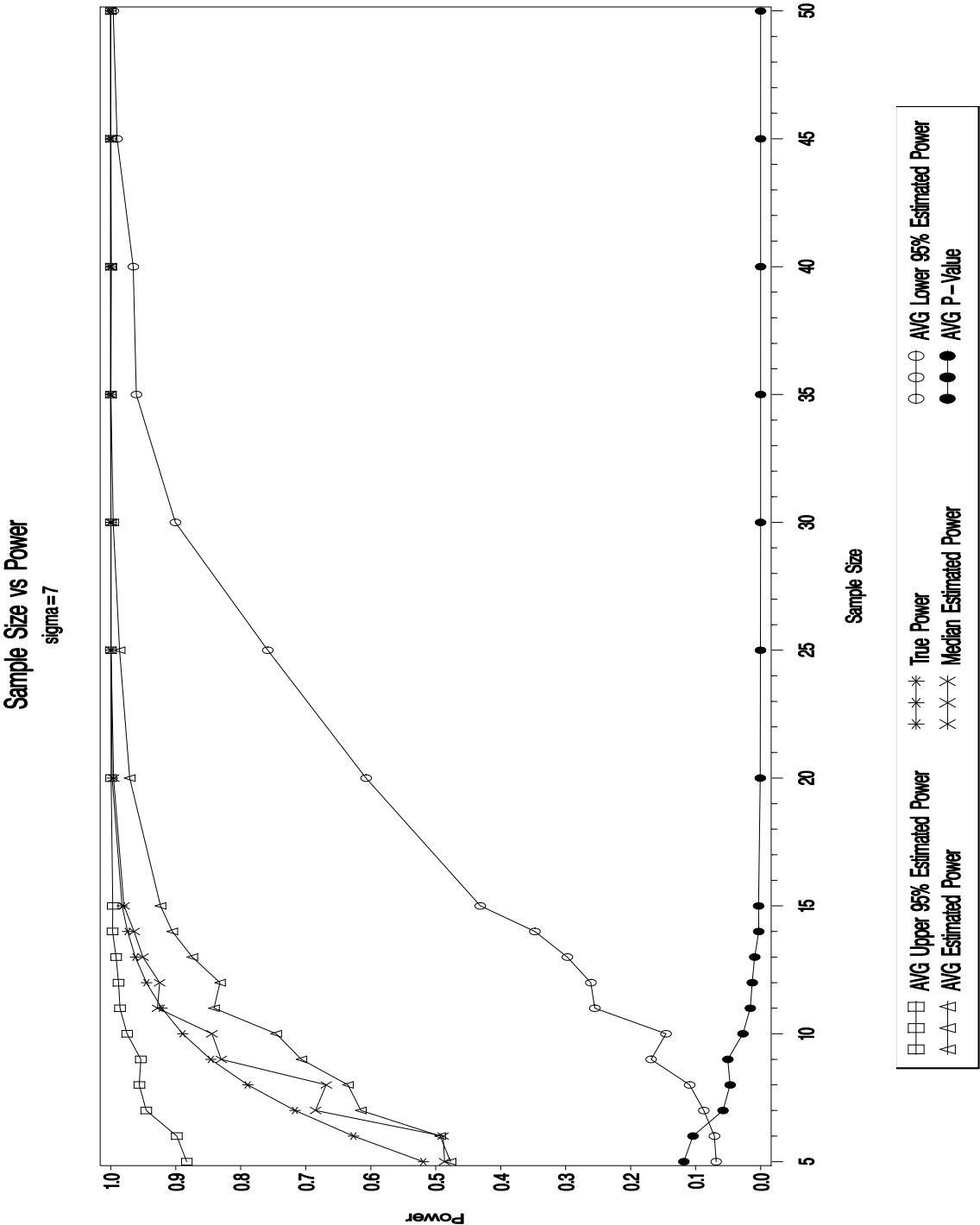


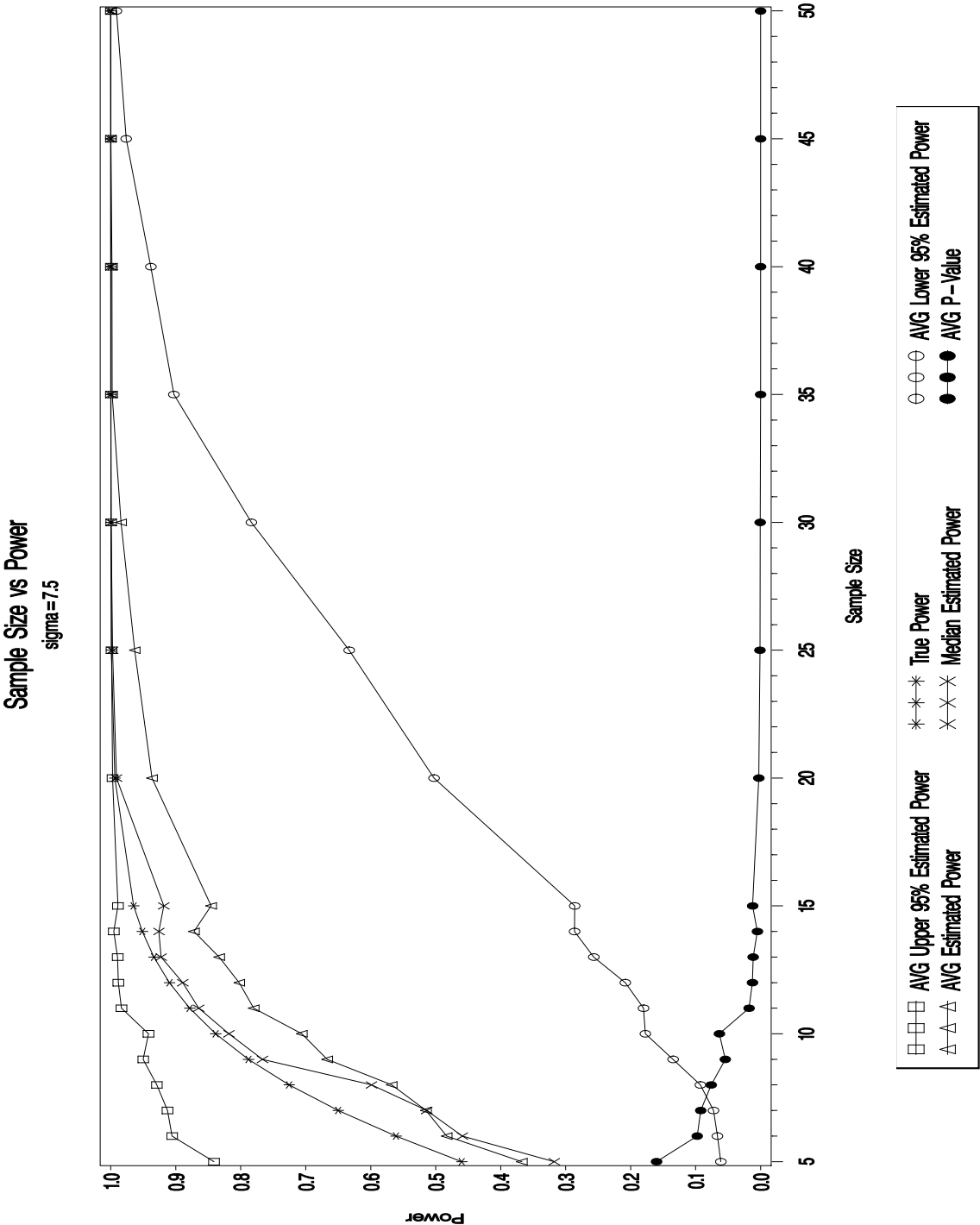




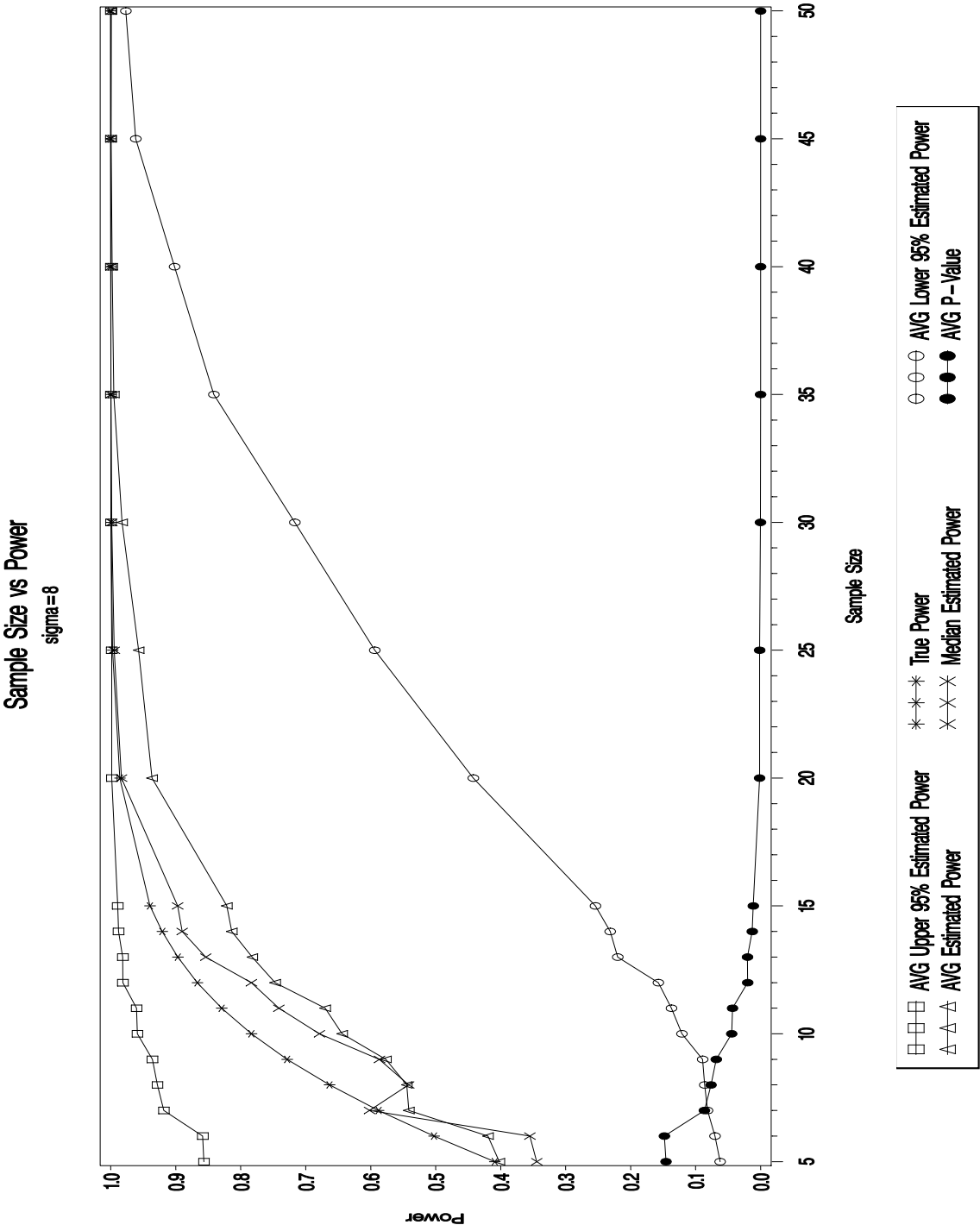


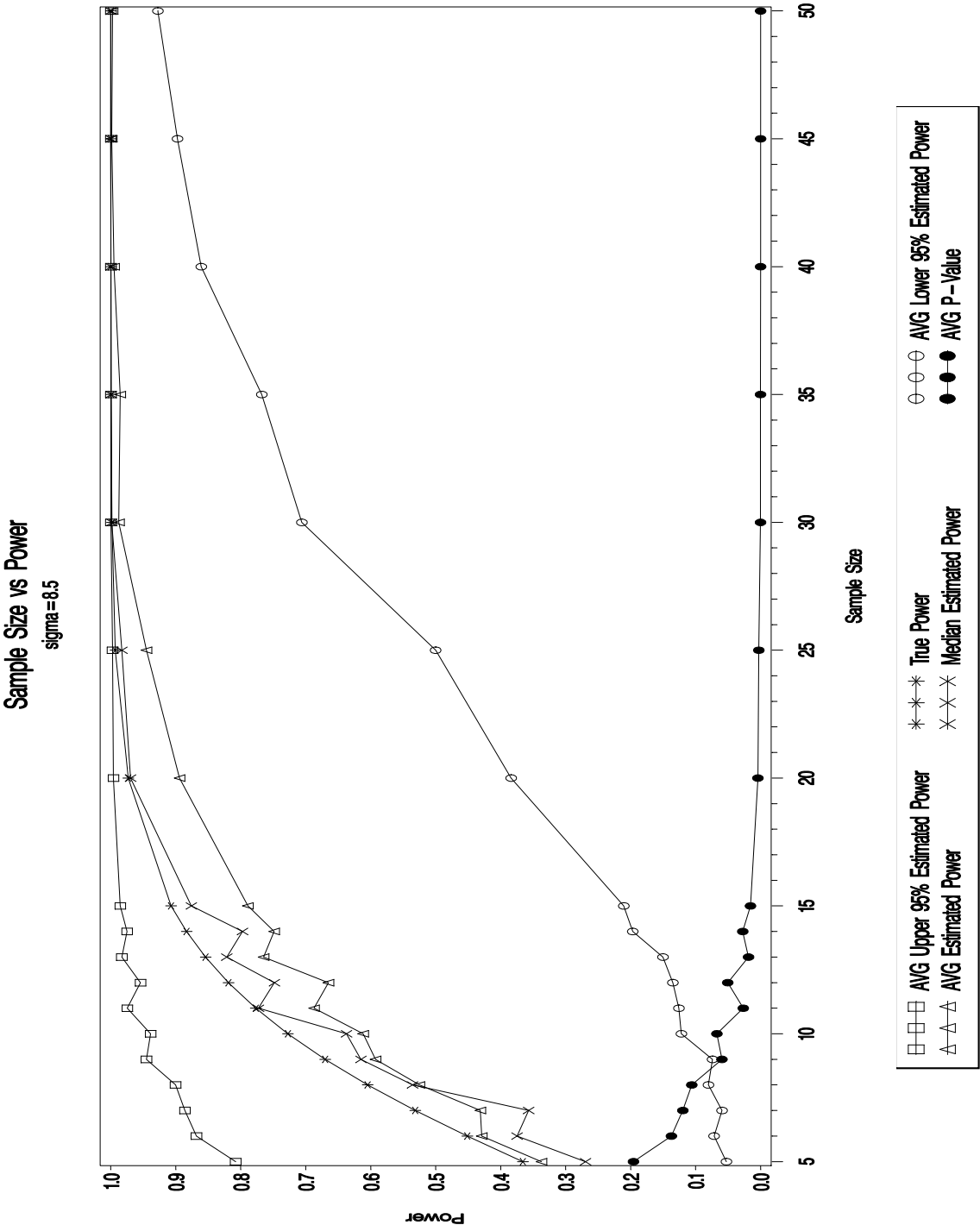


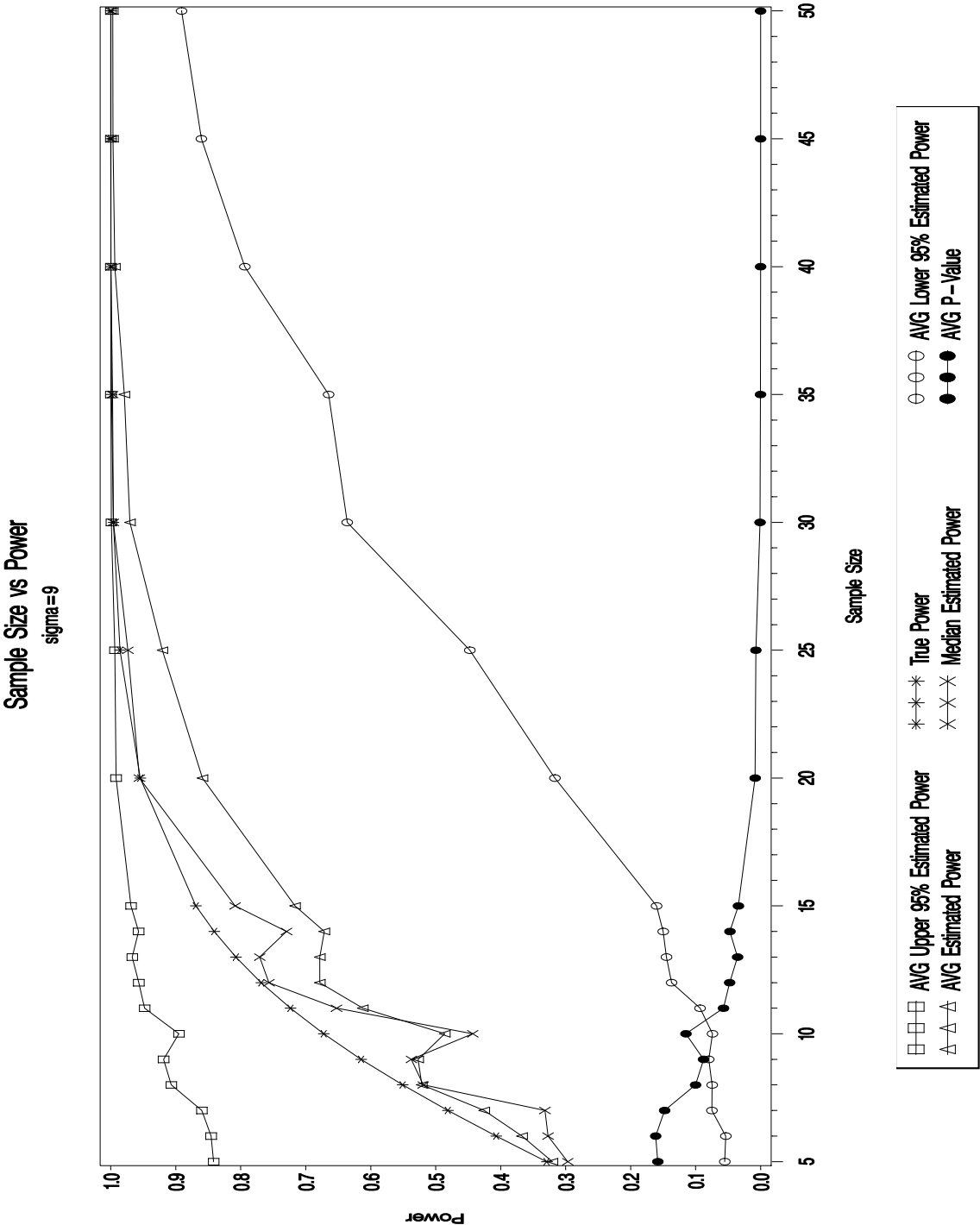


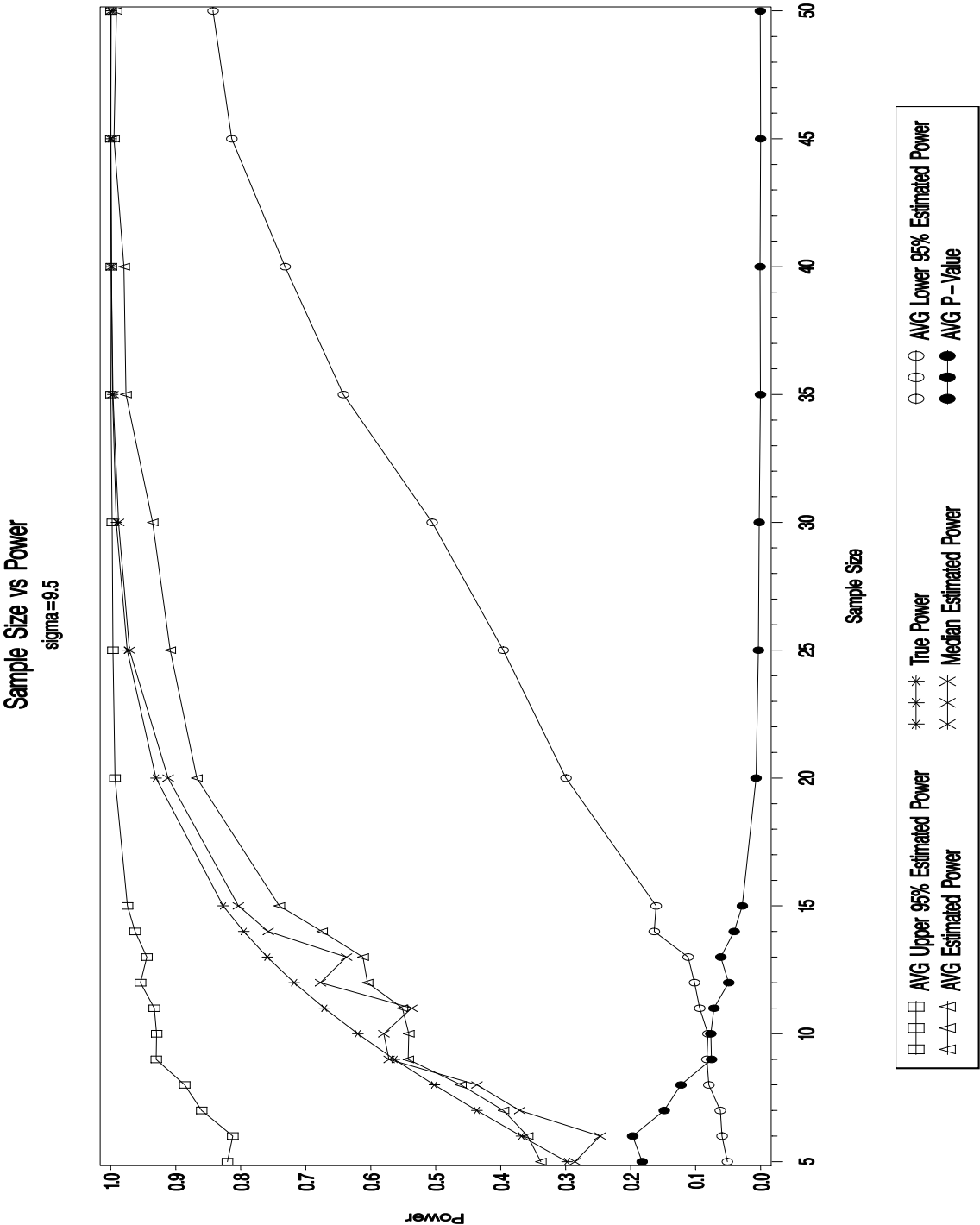


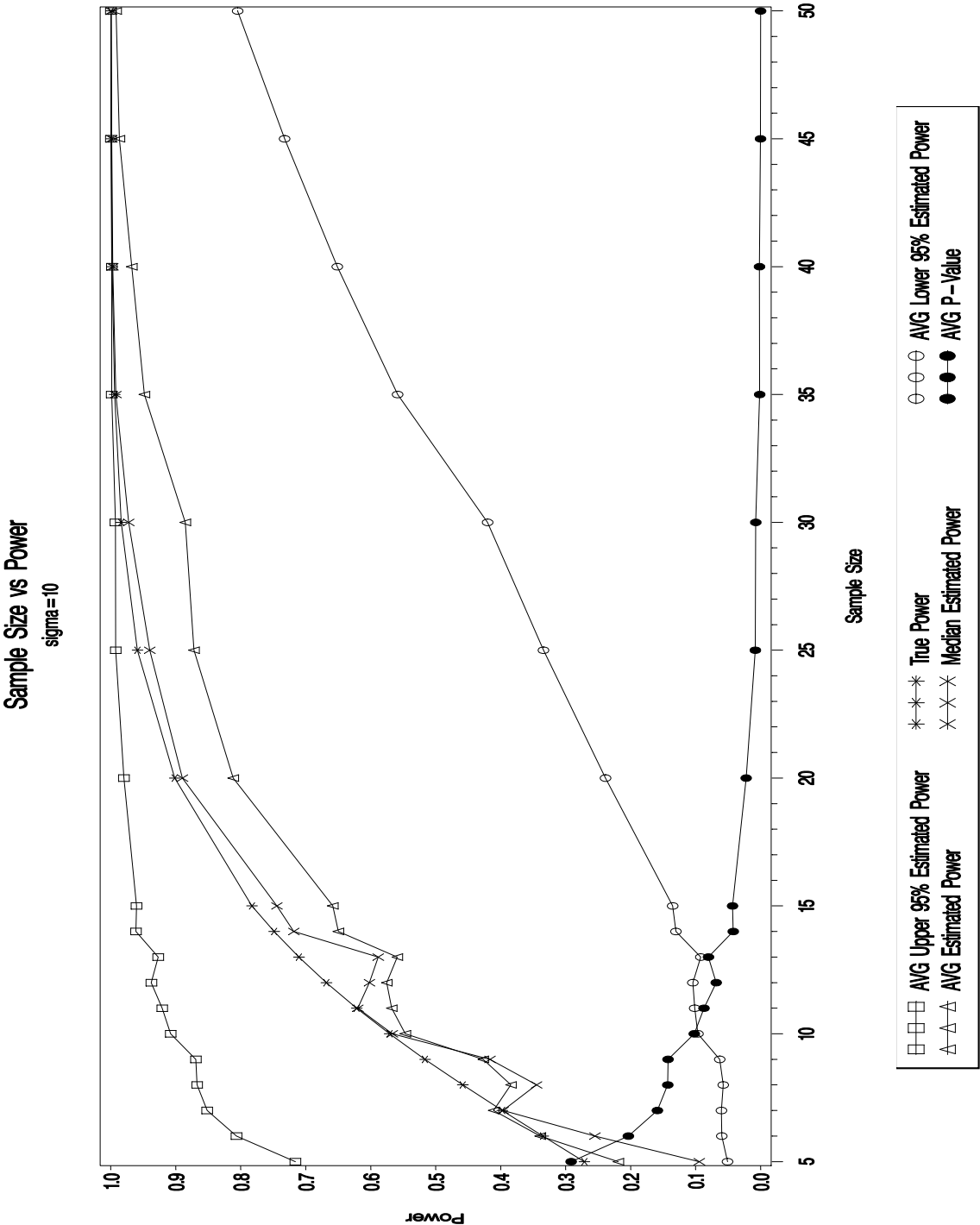


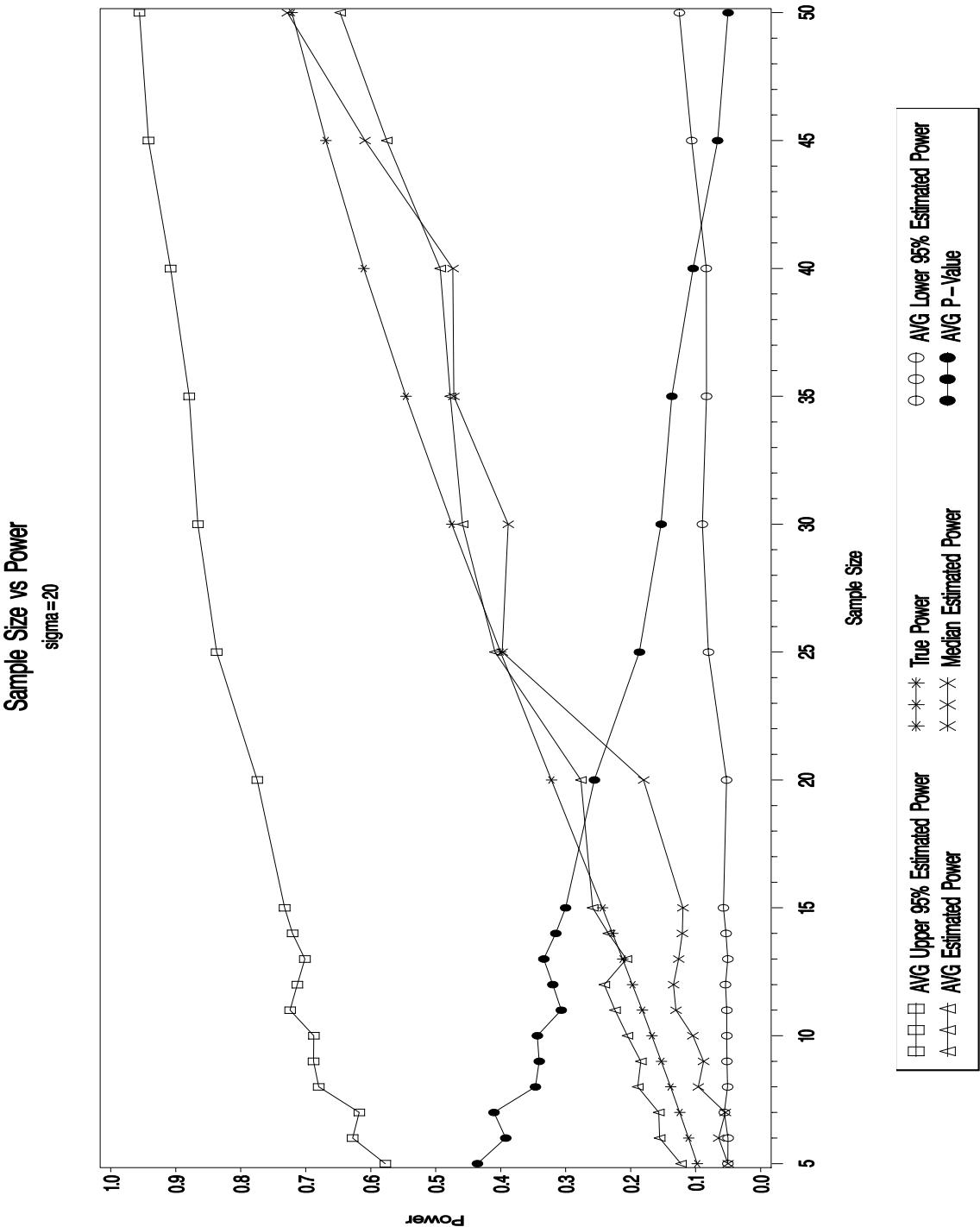












## Appendix D: SAS Code for Estimating Power for an Observed *F*-Statistic

```

1  /* Application ***** */
2  /*
3  /* Input: significance level, observed F-statistic, and
4  /* numerator and denominator degrees of freedom.
5  /*
6  /* Output: estimate for power (1-alpha)% confidence interval for power, and
7  /* p-value of the test
8  /*
9  /* ***** */
10
11 %macro application1(F=,df1=,df2=,alpha=);
12 proc iml;
13     reset nolog;
14
15     /* Function ***** */
16     /*
17     /* Return the confidence interval width for a noncentral
18     /* f-dsitribution
19     /*
20     /* ***** */
21
22     start w(alphaLow,ndf,ddf,ncent,sl);
23
24         width = quantile('F',1-sl+alphaLow,ndf,ddf,ncent)-quantile('F',alphaLow,ndf,ddf,ncent);
25         return(width);
26
27     finish w;
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

```

## Appendix D: SAS Code for Estimating Power for an Observed F-Statistic

```

48 /* Function: optimal lower critical region ***** */
49 /*
50 /* Finds the minimum confidence interval width using an
51 /* implementation of the Golden Section Search method by
52 /* Kincaid and Cheney 2009, and Jan Verscheide
53 /* http://www.math.uic.edu/~jan/mcs471/Lec9/gss.pdf
54 /*
55 /* Note(s):
56 /* The Golden Section Search method finds a value which minimizes
57 /* a function. The function must be one dimensional and unimodal.
58 /*
59 /* Imagine a 'typical' pdf of an F distribution. Say we want to
60 /* setup an interval such that the area covered by the sum of the two
61 /* tails is alpha. For each alpha we can find the
62 /* associated quantile using the SAS function 'quantile'.
63 /* Since alpha is specified, knowing the lower area easily leads to
64 /* the upper area. Here we consider only the lower
65 /* area. As such we can relate interval width as a function of
66 /* lower alpha size. As we increase or decrease the size of the
67 /* lower area we increase or decrease the width of the interval.
68 /* Assuming a minimum interval width exists, we optimize over the
69 /* values [0.00, 0.05].
70 /*
71 /* ***** */
72
73 start alphaLU(ndf1,ddf2,ncp,s1Alpha,tol);
74
75     alphaA = 0;
76     alphaB = s1Alpha;
77
78     /* Initial lower bracket on optimal region */
79     /* Initial upper bracket on optimal region */
80
81     /* Note 1 ***** */
82     /*
83     /* For each iteration an interval
84     /* [alphaA, alphaB] is retained
85     /*
86     /* ***** */
87
88     /* Golden ratio constant reduction factor */
89     /* Set test bracket a1 */
90     /* Calculate Interval width for a1 */
91     /* Set test bracket a2 */
92     /* Calculate interval width for a2 */

```



# Appendix D: SAS Code for Estimating Power for an Observed F-Statistic

```

95  /* Note 2 ***** */
96  /* */
97  /* The intial setup invokes two function */
98  /* calls. The following loop uses one */
99  /* function call per iteration. */
100 /* */
101 /* ***** */
102
103 do while (abs(alphaB-alphaA)>tol);
104
105 /* Note 3 ***** */
106 /* */
107 /* We continue until the absolute difference */
108 /* between an upper and a lower bracket is */
109 /* negligible. Tolerance is set tole=4. */
110 /* ***** */
111
112 /* Note 4 ***** */
113 /* */
114 /* Four brackets are maintained at any */
115 /* given time. */
116 /* ***** */
117
118 if wal>wa2 then do;
119
120     /* If wal > wa2 then minimum width is between */
121     /* alphaA and a1 */
122
123     /* Note 5 ***** */
124     /* */
125     /* Within the interval [alphaA, a1] we have */
126     /* the width wa2. a2 is situated by the */
127     /* expression alphaA+phi*(a1-alphaA). */
128     /* ***** */
129
130     alphaB = a1;
131     a1 = a2;
132     wa1 = wa2;
133     a2 = alphaA+phi*phi*(alphaB-alphaA);
134     wa2 = w(a2,ndf1,ddf2,ncp,slAlpha);
135 end;
136 else do;
137
138     /* If wal < wa2 then minimum width is between */
139     /* a2 and alphaB */
140
141

```

# Appendix D: SAS Code for Estimating Power for an Observed F-Statistic

```

142 /* Note 6 ***** */
143 /*
144 /* Within the interval [a2, alphaB] we have
145 /* the width wal. al is situated by the
146 /* expression alphaA+phi*phi*(alphaB-a2).
147 /*
148 /* ***** */
149
150
151 /* Updated alphaA */
152 /* Save al */
153 /* Save wal */
154 /* Set new test bracket al */
155 /* Calculate interval width for new al */
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188

```

```

/* Note 6 ***** */
/*
/* Within the interval [a2, alphaB] we have
/* the width wal. al is situated by the
/* expression alphaA+phi*phi*(alphaB-a2).
/*
/* ***** */

/* Updated alphaA */
/* Save al */
/* Save wal */
/* Set new test bracket al */
/* Calculate interval width for new al */

alphaA = a2;
a2 = al;
wal = wal;
al = alphaA+phi*(alphaB-alphaA);
wal = w(al,ndf1,ddf2,ncp,s1alpha);
end;
end;

out = (alphaA+alphaB)/2;
return(out);

finish alphaLU;

/* Calculations */

eps = 1e-4;

fcrit = quantile('F',1-&alpha,&df1,&df2);

pvalue = 1-cdf('F',&F,&df1,&df2);

estLambda = &F*&df1;
adjEstLambda = (estLambda*(&df2-2)/&df2)-&df1;

if adjEstLambda < 0 then adjEstLambda = 0;

/* Note 7 ***** */
/*
/* For some random seeds the
/* noncentrality parameter is
/* negative.
/*
/* A negative noncentrality
/* parameter occurs when
/* the expression
/* estLambda*(df2-2)/df2 is
/* less than df1.
/*
/* ***** */

```

## Appendix D: SAS Code for Estimating Power for an Observed F-Statistic

```

189
190
191 alphaL = alphaU(&df1,&df2,adjEstLambda,&alpha,eps); /* Lower significance */
192 alphaU = &alpha-alphaL; /* Upper significance */
193
194 adjAlphaL = 1-alphaL;
195 probCenF = cdf('F',&F,&df1,&df2);
196
197 /* Note 8 **** */
198 /* fnonct reports an error for a supplied */
199 /* probability greater than the */
200 /* probability from the cdf of a central */
201 /* F distribution. */
202 /*
203 /* As the probability supplied gets large */
204 /* fnonct tends toward zero. probCenF */
205 /* with fnonct is small around zero. */
206 /* Using adjAlphaL would then be smaller */
207 /* about zero. Set to zero. */
208 /*
209 /* **** */
210
211 if adjAlphaL > probCenF then lambdaL = 0;
212 else lambdaL = fnonct(&F,&df1,&df2,adjAlphaL);
213 if alphaU > probCenF then lambdaU = 0;
214 else lambdaU = fnonct(&F,&df1,&df2,alphaU);
215
216
217 estPower = 1-cdf('F',fCrit,&df1,&df2,adjEstLambda);
218 powerL = 1-cdf('F',fCrit,&df1,&df2,lambdaL);
219 powerU = 1-cdf('F',fCrit,&df1,&df2,lambdaU);
220
221 print estPower;
222 print powerL, powerU, pValue;
223
224 quit;
225 %mend application1;
226 %application1(F=1.5,df1=4,df2=10,alpha=0.05);
227

```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

```

The SAS System
15:16 Thursday, November 10, 2005 80

ESTPOWER
0.0975334

POWERL
0.05
POWERU
0.8017428
PVALUE
0.2278353

```

## Appendix D: Section 5 Output for Example 2

1	The SAS System	15:16 Thursday, November 10, 2005	81
2			
3			
4	ESTPOWER		
5	0.0777412		
6			
7			
8	POWERL		
9			
10	0.05		
11	POWERU		
12			
13	0.6713596		
14	PVALUE		
15			
16	0.2741814		
17			